ULTRIX

Guide to Configuration File Maintenance

Order Number: AA-ME90C-TE
November 1990

Product Version: ULTRIX Version 4.1 or higher

digital equipment corporation
maynard, massachusetts

Contents

Examples

Tables

About This Manual

This guide provides information on how to maintain the system configuration file and how to build a new kernel system image. This guide also explains how to build a new kernel automatically or manually.

Audience

The *Guide to Configuration File Maintenance* is written for the person responsible for managing and maintaining an ULTRIX system. It assumes that this individual is familiar with ULTRIX commands, the system configuration, the system's controller and drive unit number assignments and naming conventions, and an editor such as `vi` or `ed`. You do not need to be a programmer to use this guide.

Organization

This manual consists of two chapters, one appendix, and an index:

Chapter 1: Configuration Files
Explains the content and format of the configuration files and provides sample generic configuration files.

Chapter 2: Building the Kernel
Describes how to build a kernel either automatically or manually and explains how to build a new kernel after a capacity upgrade installation.

Appendix A: Device Mnemonics
Lists the supported device mnemonics and explains how to obtain detailed reference page information on devices.

Related Documents

You should have the hardware documentation for your system and peripherals.

Conventions

The following conventions are used in this manual:

#	A number sign is the default superuser prompt.
`user input`	This bold typeface is used in interactive examples to indicate typed user input.
`system output`	This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.
UPPERCASE lowercase	The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown.
rlogin	In syntax descriptions and function definitions, this typeface is used to indicate terms that you must type exactly as shown.
filename	In examples, syntax descriptions, and function definitions, italics are used to indicate variable values; and in text, to give references to other documents.
[ ]	In syntax descriptions and function definitions, brackets indicate items that are optional.
. . .	In syntax descriptions and function definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
`cat`(1)	Cross-references to the *ULTRIX Reference Pages* include the appropriate section number in parentheses. For example, a reference to `cat`(1) indicates that you can find the material on the `cat` command in Section 1 of the reference pages.

1 Configuration Files

This chapter explains the contents and format of the configuration files for VAX and RISC processors. The chapter provides a sample generic configuration file in each case to illustrate how specific information defines the hardware, software, and system parameters.

In addition to the configuration file information contained in this chapter, the following information will help you to understand the software and hardware components of your system:

- Section 4 of the *ULTRIX Reference Pages* contains definitions of supported devices, special files, interfaces, and system utilities involved in system configuration. For example, see `autoconf`(4) for a description of the configuration diagnostics utility; `rz`(4) for a description of the SCSI disk driver; `ra`(4) for a description of the MSCP disk driver; `ln`(4) for the description of the Ethernet interface; and `tz`(4) for a description of the SCSI tape driver.
- Section 8 of the *ULTRIX Reference Pages* contains definitions of various system utilities used during the configuration process. For example, see `config`(8) and `MAKEDEV`(8).
- Appendix A provides information on the MAKEDEV script and lists the names of the device mnemonics supported by MAKEDEV.

1.1 The System Configuration File

The system configuration file describes how you want the configuration software to build the kernel. It identifies all of the device driver source code that needs to be compiled into the kernel, as well as a number of system parameters that affect how the kernel operates. The kernel is the system image that controls system scheduling, memory management, input and output services, device management, and organization of the file systems. Provided you have enough disk space, you can build more than one kernel.

- For VAX processors, the system configuration file resides in `/usr/sys/conf/vax` and has the same name as the system name (in uppercase letters) that you defined during the installation procedure. For example, if you named your system `tucson` during the installation procedure, then the system configuration file name will be `/usr/sys/conf/vax/TUCSON`.
- For RISC processors, the system configuration file resides in `/usr/sys/conf/mips` and has the same name as the system name (in uppercase letters) that you defined during the installation procedure. For example, if you named your system `tucson` during the installation procedure, then the system configuration file name will be `/usr/sys/conf/mips/TUCSON`.

1.2 The Generic System Configuration File

The installation software provides you with a generic system configuration file that you can use as a template to build or tailor other configuration files.

- For VAX processors, the generic configuration file is `/usr/sys/conf/vax/GENERIC`.
- For RISC processors, the generic configuration file is `/usr/sys/conf/mips/GENERIC`.

All configuration files, including the generic configuration file, have multiple sections:

- Global definitions
- Options definitions
- Makeoptions definitions
- System image definitions
- Device definitions
- Pseudodevice definitions

Note

> Some of the preceding configuration file sections may not be relevant to your processor. For example, the makeoptions definitions apply only to RISC processors. Similarly, some of the system parameters discussed later in this section will not appear in your configuration file. These parameters, as well as some of the arguments to the parameters, are described here because they may be used in some system configuration files.

1.2.1 Global Definitions

The global definitions parameters apply to all the kernels generated by the configuration file. Each global definition appears on a separate line in the configuration file.

Each line represents a tunable system parameter and begins with one of these keywords:

```
machine
cpu
ident
timezone
maxusers
maxuprc
maxuva
physmem
bufcache
swapfrag
maxtsiz
maxdsiz
maxssiz
smmin
smmax
smseg
smsmat
```

```
smbrk
processors
scs_sysid
```

The following paragraphs display the syntax and describe how and when to use each parameter:

machine *type*

This parameter defines the hardware; the argument *type* must be `vax` for VAX machines and `mips` for RISC machines. For example, to define a VAX processor to the configuration file, enter:

```
machine vax
```

cpu "*type*"

This parameter defines the processor; the argument *type* must be enclosed in quotes. For example, to define a DECstation 3100 CPU, enter:

```
cpu "DS3100"
```

For VAX processors, the generic configuration file lists the CPU types by processor class. The configuration file lists the processors by CPU type because, in some cases, the configuration software assigns an equivalence name to the processor name. For instance, the MVAX entry applies to the MicroVAX II and VAXstation 2000 processors. The VAX3600 entry in the GENERIC configuration file applies to all of the MicroVAX 3000, VAX 3000, and VAXserver 3000 families of processors.

If you know your processor class, you can use the processor class for your configuration file entry. If you do not know your processor class, you can use the exact processor name. For example, you can use one of the following processor names:

```
DS3100
DS5000
DS5400
DS5500
DS5800
VAX8800
VAX8820
VAX8700
VAX8600
VAX8550
VAX8530
VAX8500
VAX8350
VAX8300
VAX8200
VAX6400
VAX6210
VAX6220
VAX3600
VAX3500
VAX3400
VAX3300
VAX785
VAX780
VAX750
VAX420
MVAX
```

Note

You can specify more than one `cpu` *type* entry in the configuration file for a kernel that can be booted on multiple CPUs. However, if you specify more than one cpu *type* entry, your system builds more capabilities than it needs. The result in most cases is that your kernel requires more memory than a kernel for a single processor requires. Under these conditions, your system may page and swap frequently during daily operations, which affects system performance.

ident *name*

This parameter defines the host machine for which you are creating the configuration file. The *name* argument is the system name that you specified during the installation procedure. Enter the name in uppercase letters. For example, the following defines the host machine TUCSON:

```
ident TUCSON
```

This parameter ensures that all host-specific source code is compiled during the actual configuration process.

timezone *number* dst *x*

This parameter defines timezone information for your site. The installation procedure enters this value to your system configuration file according to information you supply during the installation or when you register a diskless client. The *number* argument identifies your time zone, measured by the number of hours west of Greenwich Mean Time. For example, Eastern Standard Time is five hours west of Greenwich Mean Time, and Pacific Standard Time is eight hours west. Negative numbers indicate hours east of GMT. The generic configuration file time zone entry is set to Eastern Daylight Savings Time (the entry is `timezone 5 dst`).

The argument `dst` indicates daylight savings time. During the installation procedure, you can include a number (*x*) to request a particular daylight savings time correction algorithm. The values are as follows:

1 United States (the default value)

2 Australia

3 Western Europe

4 Central Europe

5 Eastern Europe

maxusers *number*

This parameter defines the maximum number of simultaneously active users allowed on your system. The *number* argument should be equal to or greater than the maximum number of users allowed by your license agreement.

The number in this field is used in the system algorithms to size a number of system data structures and to determine the amount of space allocated to system tables. One such table is the system process table, which is used to determine how many active processes can be running at one time.

maxuprc *number*

This parameter defines the the maximum number of processes one user can run simultaneously. The default `maxuprc` entry is 50.

maxuva *num*

This parameter defines the maximum aggregate size of user virtual address space in megabytes allowed by the system. The default value is 256 megabytes. This parameter does not apply to RISC processors.

physmem *number*

This parameter defines an estimate of the amount of physical memory currently in the system, in megabytes. This *number* argument is not used to limit the amount of memory; it is used by the system to size the system page table. Consequently, it should be greater than or equal to the amount of physical memory in the system.

bufcache *percent*

This parameter defines the amount of physical memory that is to be allocated for use by the file system buffer cache. The *percent* argument must be greater than or equal to 10 but less than 100. The specified percentage of the actual amount of physical memory found at boot time is allocated for this purpose; this memory is never used for other purposes.

At boot time, if there is not enough memory to satisfy minimum system needs, the percentage is automatically reduced and a diagnostic message is issued.

Because the buffer cache places a demand on the system page table, correct operation requires a sensible value for `physmem` as described previously. At boot time, if the system page table is too small to support the buffer cache, this percentage is automatically reduced and a diagnostic message is issued.

Note

The amount of physical memory used for the buffer cache is never used for program execution. A high percentage may help data intensive applications but cause problems with applications that require the majority of memory for program execution.

The following example shows the format of the buffer cache parameter:

```
bufcache            25
```

swapfrag *number*

The system satisfies requests for additional swap space using the value `swapfrag`. A process is granted *number* 512-byte blocks of swap space each time the process requests swap space.

When the `swapfrag` number increases, the swap space wastage also increases. The minimum value of *number* should be at least 16. The default value of *number* is 64. The *number* value must be a power of 2.

maxtsiz *num*

This parameter defines the largest text segment in megabytes allowed by the system.

- For VAX processors, the default value is 12 megabytes.
- For RISC processors, the default value is 32 megabytes.

maxdsiz *num*

This parameter defines the largest data segment, in megabytes, allowed by the

system. The default value is 32 megabytes.

Note

You must use `maxdsiz` to increase the data segment size, and `maxsiz` to increase the stack segment size. The parameters `dmmax` and `dmmin` are no longer supported.

maxssiz *num*

This parameter defines the largest stack segment in megabytes allowed by the system. The default value is 32 megabytes.

smmin *num*

- For VAX processors, this parameter defines the minimum number of 512-byte pages of virtual memory at which a shared memory segment (SMS) may be sized. The default for `smmin` is 0 blocks.
- For RISC processors, this parameter defines the minimum number of 4096-byte pages of virtual memory at which a shared memory segment (SMS) may be sized. The default for `smmin` is 0 pages.

For more information, see `shmget`(2) in the *ULTRIX Reference Pages.*

smmax *num*

- For VAX processors, this parameter defines the maximum number of 512-byte pages of virtual memory at which a shared memory segment may be sized. The default for `smmax` is 256 blocks (128 Kbytes).
- For RISC processors, this parameter defines the maximum number of 4096-byte pages of virtual memory at which a shared memory segment may be sized. The default for `smmax` is 32 pages (128 Kbytes).

For more information, see `shmget`(2) in the *ULTRIX Reference Pages.*

smseg *num*

This parameter defines the maximum number of shared memory segments per process. The default value is 6. For more information, see `shmop`(2) in the *ULTRIX Reference Pages.*

smsmat *num*

This parameter defines the highest attachable address, in megabytes, for shared memory segments.

- For VAX processors, the default value is MAXDSIZE.
- For RISC processors, the default value is 0. Although the parameter is valid, be aware that this check is not made.

For more information, see `shmop`(2) in the *ULTRIX Reference Pages.*

smbrk *num*

- For VAX processors, this parameter defines the default spacing between the end of a private data space of a process and the beginning of its shared data space in 512-byte pages of virtual memory. This value is important, because once a process attaches shared memory, private data cannot grow past the beginning of shared data. The default for `smbrk` is 64 pages (32 Kbytes).

- For RISC processors, this parameter defines the default spacing between the end of a private data space of a process and the beginning of its shared data space in 4096-byte pages of virtual memory. This value is important because, once a process attaches shared memory, private data cannot grow past the beginning of shared data. The default for `smbrk` is 10 pages (40 Kbytes).

For more information on shared memory operations, see `shmop`(2) in the *ULTRIX Reference Pages.*

processors *num*

This parameter defines the number of processors in the system.

scs_sysid *number*

This parameter identifies each host uniquely on the CI star cluster to the SCS subsystem. The *number* argument must be a unique identifier for each host. At installation, the system automatically generates this number and puts it in the configuration file. If the system does not detect a CI at installation, it provides a default value of 1.

1.2.2 Options Definitions

The options definitions parameters specify optional code to be compiled into the system. You should leave the options as they appear in the generic configuration file. However, you can remove any of the options if they do not pertain to your site or if your system is short on physical memory space.

The syntax for options definitions is:

`options` *optionlist*

The possible values for *optionlist* are:

EMULFLT

This option enables emulation of the floating point instruction set if it is not already present in the hardware.

FULLDUMPS

This option enables full dump support.

INET

This option provides internet communication protocols. The `inet` pseudodevice must also be set.

LAT

This option allows you to access your machine from a local area terminal server on the Ethernet. The `lta` and `lat` pseudodevices must also be set.

DECNET

If the DECnet layered product is installed, this option must be set. The `decnet` pseudodevice must also be set.

QUOTA

This option allows disk quotas to be set.

SYS_TRACE

This option enables the system call tracing capability. The `sys_trace` pseudodevice must also be set.

DLI

This option allows the mop_mom program to be active. The mop_mom command is usually included in the /etc/rc.local file as a background task to cause mop_mom to listen for down-line and up-line load requests over the network. The dli pseudodevice must also be set.

SYS_TPATH

This option enables the trusted path mechanism. The sys_tpath pseudodevice must also be set.

RPC

This option allows RPC-based applications. It is required for the NFS option. The rpc pseudodevice must also be set.

NFS

This option allows you to access the NFS protocol. It requires both the RPC option and the nfs pseudodevice to be set.

UFS

This option enables the standard, local file system. If you do not use the NFS option, the UFS option must be set. If you do not specify this option, the system will be considered diskless. The ufs pseudodevice must also be set.

AUDIT [= *number*]

This option loads the optional audit subsystem files into the kernel. To specify the base size of the audit buffers in bytes, use the *number* option. The default base size of the audit buffers is 16K bytes.

SMP

This option allows multiples processors to run. If this option is set on a single processor, there is a performance penalty. This option should not be used with a single processor.

1.2.3 The makeoptions Definitions for RISC Processors

You can specify one makeoptions definition in the generic configuration file for RISC processors. The format of the makeoptions definition is as follows:

makeoptions *OPTION_NAME*="*argument*"

The *OPTION_NAME* variable must be in uppercase letters. The *argument* variable must be placed within quotation marks ("). The *OPTION_NAME* and *argument* variables are separated by an equal sign (=). The makeoptions definition follows:

ENDIAN="–EL"

This definition specifies the byte order within words used by the processor, and must be "–EL."

1.2.4 System Image Definitions

There is one system definition in the generic configuration file. However, you can change the definition or add more lines to the configuration file you are building to indicate that you want to generate more than one kernel. For each kernel you wish to generate, specify one line that begins with the keyword config. Each line can be used to define the root device, the swap area or areas, the dump area, and the argument processing area for system calls.

The general format for the system image definition is as follows:

`config` *filename configuration-clauses*

The *filename* argument is the name to be assigned to the file constituting the compiled kernel, or system image. The installation procedure assigns the name `vmunix`. The *configuration-clauses* define the devices for the root file system, for the paging and swapping area, and for crash dumps. The *configuration-clauses* keywords are `root`, `swap`, and `dumps`. The syntax and descriptions of these keywords are as follows:

root [on] *device*

The installation procedure assigns partition `a` of the system disk to the root file system. You can change this assignment by editing the configuration file. For diskless clients, this entry is set to `root on ln0`.

Some configuration file entries for the system image definition are as follows:

```
config    vmunix    root on ln0
config    vmunixa   root on rz0a
```

The first entry specifies that the root file system resides on the remote (network) boot device. You must use this entry for diskless clients. The second entry specifies that the root file system resides on partition `a` of the local boot device, rz drive 0.

swap [on] *device* [and *device*] [size *x*] [boot]

The first *device* argument specifies the device and partition that you want the system to use for a paging and swapping area. The installation procedure assigns partition `b` of the system disk for the paging and swapping area. You can change this assignment by editing the configuration file.

The second *device* argument enables you to add another partition, so the kernel interleaves paging and swapping between the two partitions. To specify a second paging and swapping area, use the `and` clause with a device, a logical unit, and a partition name.

Use the `size` clause to specify a nonstandard partition size for one or more swap areas. The value of *x* represents the number of 512-byte sectors. A size larger than the associated disk partition is trimmed to the partition size. The default swap device is partition `b` of the device where the root is located.

If you specify `swap on boot`, the `a` partition of the booted device becomes the root, and swap space is assumed to be the `b` partition of the same device.

Example configuration file entries are as follows:

```
config   vmunix    swap on boot
config   vmunixa   root on ln0 swap on rz0b
config   vmunixb   root on rz0a swap on rz0b
```

In the first example, the root file system is on partition `a` of the local boot device, and partition `b` of the same device becomes the swap space. In the second example, the root file system resides on the remote (network) boot device, but the system swaps on partition `b` of the local disk at drive 0. In the last example, the root file system is on partition `a` of the local boot device (rz0), and the system swaps on partition `b` of the same device.

You can also swap between two disks. For example, if you specify a swap on both rz0b and rz1b, the system can swap on partition `b` of either disks. Note that you must add an entry to the `/etc/fstab` file that specifies mounting of the second disk.

For diskless systems, if the swap file is remote, then you do not have to specify a swap device.

Avoid selecting partition `a` of any disk for use as the swap partition. If partition table information was defined for a disk and swapping occurs on the `a` partition, the information is destroyed and data is lost.

dumps [on] *device*

The *device* argument specifies the partition and the device where crash dumps are to be stored. The device that is specified must be on the same controller as the boot device. The default dump device is the first swap device configured.

Usually, this entry is unnecessary in a diskless environment, because the `dms` setup process specifies using the `mop_mom` command for dumping. For a description of this command, see `mop_mom(8)` in the *ULTRIX Reference Pages*. For more information on diskless environments, see the *Guide to Diskless Management Services*.

1.2.5 Device Definitions

This section of the configuration file contains descriptions of each current or planned device on the system. You need to add definitions for devices that were not on the system at installation time. You may also want to delete device definitions for devices that have been removed from the hardware configuration.

Each line of this section of the file begins with one of these keywords:

adapter	Identifies a physical connection to a system bus such as VAXBI, MASSBUS, Q–bus, UNIBUS, MSI, IBUS, or CI.
master	A MASSBUS tape controller.
controller	Identifies either a physical or a logical connection with one or more slaves attached to it. Some examples are uda, kdb, hsc, and uq.
device	An autonomous device which connects directly to a Q–bus, or to a UNIBUS, MASSBUS, IBUS, or VAXBI adapter (as opposed to a disk, for example, which connects through a disk controller).
disk	A disk drive connected to either a master or a controller.
tape	A tape drive connected to either a master or a controller.

The format of the information required for each of these types of devices varies, as described in the following sections.

1.2.5.1 Adapter Specifications – The adapters discussed in this section are the VAXBI, MASSBUS, UNIBUS, MSI, CI, IBUS, and Q-bus adapters. Each adapter is specified by its own format in the configuration file.

The format for VAXBI adapters is:

adapter vaxbi*n* at nexus*?*

The *n* is the unit number of the adapter. The question mark (*?*) allows the system to pick the appropriate NEXUS for you.

The format for MASSBUS adapters is:

adapter mba*n* at nexus*?*

The *n* is the unit number of the adapter. The question mark (*?*) allows the system to pick the appropriate NEXUS for you.

The format for IBUS adapters is:

adapter ibus*n* at nexus*?*

The *n* is the unit number of the adapter. The question mark (*?*) allows the system to pick the appropriate NEXUS for you.

The format for UNIBUS and Q-bus adapters is the same. Q-bus adapters are specific to MicroVAX-type and VAXstation-type processors. The format is:

adapter uba0 at nexus*?*

The question mark (*?*) allows the system to pick the appropriate NEXUS for you.

The format for MSI adapters is:

adapter msi0 at nexus*?*

The question mark (*?*) allows the system to pick the appropriate NEXUS for you.

The formats for CI adapters are:

adapter ci0 at nexus*?*
adapter ci0 at vaxbi*?*

The question mark (*?*) allows the system to pick the appropriate NEXUS or VAXBI for you.

1.2.5.2 Master Specifications – MASSBUS tape drives must be attached to a master.

The format for specifying a master is:

master *dev*name at mba*m* drive*n*

dev The name of the tape device, such as ht0.

m The MASSBUS adapter number.

n The drive number.

For example:

```
master      ht0      at mba?      drive?
tape        tu0      at ht0       slave 0
tape        tu1      at ht0       slave 1
```

1.2.5.3 Controller Specifications – This section contains examples of the specifications for the various controllers.

The controller examples are for MSCP, TMSCP, and SCSI controllers. This section also defines the format for specifying other tape-to-disk interface controllers.

The specifications for MSCP disk controllers are as follows:

- For UNIBUS or Q-bus:

```
controller  uda0 at uba0
controller  uq0 at uda0 csr 0172150 vector uqintr
disk ra0 at uq0 drive 0
disk ra1 at uq0 drive 1
disk ra2 at uq0 drive 2
disk ra3 at uq0 drive 3
```

- For VAXBI:

```
controller kdb0 at vaxbi0 node?
controller uq0 at kdb0 vector uqintr
disk ra0 at uq0 drive 0
disk ra1 at uq0 drive 1
disk ra2 at uq0 drive 2
disk ra3 at uq0 drive 3
controller aio1 at vaxbi? node?
controller bvpssp0 at aio1 vector bvpsspintr
disk ra0 at bvpssp0 drive 0
```

- For VAX CI/HSC:

```
adapter ci0 at nexus?
adapter ci0 at vaxbi? node?
controller hsc0 at ci0 cinode0
disk ra0 at hsc0 drive0
```

- For MSI bus:

```
adapter msi0 at nexus?
controller dssc0 at msi0 msinode 0
disk ra0 at dssc0 drive 0
```

- For XMI:

```
controller kdm0 at xmi0 node?
controller uq0 at kdm0 vector uqintr
disk ra0 at uq0 drive 0
disk ra1 at uq0 drive 1
disk ra2 at uq0 drive 2
disk ra3 at uq0 drive 3
disk ra4 at uq0 drive 4
disk ra5 at uq0 drive 5
disk ra6 at uq0 drive 6
disk ra7 at uq0 drive 7
```

The specifications for TMSCP tapes controllers are as follows:

- For UNIBUS or Q-bus:

```
controller klesiu0 at uba0
controller uq0 at klesiu0 csr 0174500 vector uqintr
tape tms0 at uq0 drive 0
```

- For VAXBI:

```
controller klesib0 at vaxbi0 node 0
controller uq0 at klesib0 vector uqintr
tape tms0 at uq0 drive 0
controller aie0 at vaxbi? node?
controller bvpssp0 at aie0 vector bvpsspintr
tape tms0 at bvpssp0 drive 0
```

- For MSI Bus:

```
adapter msi0 at nexus?
controller dssc0 at msi0 msinode0
tape tms0 at dssc0 drive 0
```

- For VAX CI/HSC:

```
adapter ci0 at nexus?
adapter ci0 at vaxbi? node?
controller hsc0 at ci0 cinode0
tape tms0 at hsc0 drive 0
```

- For XMI:

```
controller kdm0 at xmi0 node?
controller uq0 at kdm0 vector uqintr
tape tms0 at uq0 drive 0
tape tms1 at uq0 drive 1
```

There are three types of SCSI controllers: scsi, sii, and asc. The generic specifications for SCSI controllers for both tape and disks are as follows:

- For disks:

```
adapter      uba0 at nexus?
controller  scsi0 at uba0  csr 0x200c0080 vector szintr
controller  scsi0 at uba0  csr 0x200c0080 vector szintr
disk  rz1 at scsi0   drive 1
disk  rz2 at scsi0   drive 2
disk  rz9 at scsi1   drive 1
disk  rz10 at scsi1  drive 2

controller  sii0 at ibus?  vector sii_intr
disk  rz0 at sii0   drive 0
disk  rz1 at sii0   drive 1
disk  rz2 at sii0   drive 2
disk  rz3 at sii0   drive 3
disk  rz4 at sii0   drive 4

controller asc0 at ibus?  vector ascintr
controller asc1 at ibus?  vector ascintr
controller asc2 at ibus?  vector ascintr
disk  rz1 at asc0  drive 1
disk  rz2 at asc0  drive 2
disk  rz9 at asc1  drive 1
disk  rz13 at asc1  drive 5
disk  rz17 at asc2 drive 1
disk  rz20 at asc2  drive 4
```

- For tapes:

```
adapter      uba0 at nexus?
controller  scsi0 at uba0  csr 0x200c0080 vector szintr
controller  scsi0 at uba0  csr 0x200c0080 vector szintr
tape  tz1 at scsi0   drive 1
tape  tz2 at scsi0   drive 2
tape  tz9 at scsi1   drive 1
tape  tz10 at scsi1  drive 2

controller  sii0 at ibus?  vector sii_intr
tape  tz0 at sii0   drive 0
tape  tz1 at sii0   drive 1
tape  tz2 at sii0   drive 2
```

```
controller asc0 at ibus?  vector ascintr
controller asc1 at ibus?  vector ascintr
controller asc2 at ibus?  vector ascintr
tape  tz1 at asc0  drive 1
tape  tz2 at asc0  drive 2
tape  tz9 at asc1  drive 1
tape  tz13 at asc1  drive 5
tape  tz17 at asc2 drive 1
tape  tz20 at asc2  drive 4
```

The following specification describes the format for the magnetic tape interface (ts) and the disk interface:

controller *dev* at *condev* [csr *n*] vector *vec*
tape *unit* at *dev* drive *n*

dev — The device name and logical unit number of the controller.

condev — The name and logical unit number of the device to which the controller is connected.

n — For the controller, *n* represents the 16-bit octal address of the control status register for the device. This entry is not needed for the VAXBI. For the tape, *n* represents the logical name of the tape unit.

unit — The unit number of the tape drive.

vec — The address of any interrupt vector for the controller.

This example shows a sample entry for a TU80 or TSV05 (for MicroVAX systems) magnetic tape interface:

```
controller zs0 at uba0 csr 0172520 vector tsintr
tape       ts0 at zs0 drive 0
```

1.2.5.4 Device Specifications – The format for the hardware classified as devices is:

device *dev* *condev* [csr *n*] [flags *f*] vector *v1* ...

Tab characters are used to indicate continuation lines, if needed. The arguments are:

dev — The device name and logical unit number of the device.

condev — The name and logical unit number of the adapter or controller to which the device is connected.

n — The octal address of the control status register for the device. The `csr` *n* option is not needed for VAXBI devices. A number used to convey information about the device to the device driver. The only flags for Digital-supported devices are for line printers and communications multiplexers.

f — The default page width for all Digital line printers is 132 columns. To change the page width, use `flags` *f*, where *f* is a decimal number giving the desired width in columns. For example, to change to 80 columns, enter `flags 80`.

The DH, DZ, DMB, DHU, DMF, and DMZ communications multiplexers accept a hexadecimal flag value to specify any lines that should be treated as hardwired, with carrier always present. The DHV–11, DZQ, and DZV serve the same function as the Q-bus. The format of the hexadecimal number is 0x*nn*, where *nn* is a hexadecimal

number consisting of digits ranging from 0-9, a-f.

Because bits are numbered from right to left, setting bit 0 of the flag indicates that tty00 is hardwired, setting bit 1 of the flag indicates that tty01 is hardwired, and so forth. This example shows that tty02 is hardwired with carrier always present: `flags 0x04`.

v1... The names of interrupt vector routines for the device driver.

The following example shows a sample device specification for the DEUNA 10-Mbyte Ethernet interface:

```
device de0 at uba0 csr 0174510 vector deintr
```

The following example shows a sample device specification for a DZ–11 communications multiplexer:

```
device dz0 at uba0 csr 0160100 flags 0xff vector dzrint dzxint
```

The following example shows a sample device specification for a DMB32 communications controller device:

```
device dmb0 at vaxbi2 node3 flags 0x00ff vector dmbsint dmbaint dmblint
```

1.2.5.5 Disk Specifications – The format for specifying disks is:

disk *dev* at *condev* drive *n*

dev The device name and logical unit number of the disk.

condev The name and logical unit number of the adapter or controller to which the disk is connected.

n The physical unit number of the disk. If your disk is an MSCP (RA) unit, or if your disk is on a MASSBUS device, you can specify a question mark (?) for *n*. A question mark (?) allows the system to assign the physical number to the disk for you.

Here is an example of a device specification for MSCP disks:

```
disk ra0 at uq0 drive 0
```

1.2.6 Pseudodevice Definitions

A pseudodevice is an operating system component for which there is no associated hardware; for example, a pseudoterminal or one of the various supported protocols. The configuration file contains pseudodevice definitions to allow the operating system to recognize these components.

Each pseudodevice definition line in the configuration file defines a driver for a particular pseudodevice. Each pseudodevice definition line begins with the keyword `pseudodevice`, followed by the pseudodevice name. The format is:

pseudo-device *name* [*num*]

The *name* variable defines the name of the pseudodevice. The *num* argument specifies a number that is different from the default value.

The possible values for *name* and *num* are:

pty — Pseudoterminal support. The default is 32. Specify *num* if more than 32 pseudoterminals are defined in your configuration file. For example, to assign 64 pseudoterminals, specify `pseudo-device pty 64` in increments of 16.

inet — DARPA internet protocols.

loop — Network loopback interface.

presto — Enables kernel support for the ULTRIX Prestoserve product on the DS5500. This pseudodevice is automatically placed in the configuration file during the installation procedure.

ether — 10-Mbyte Ethernets.

lat — Local area terminal (LAT) protocols; must include `lta`.

lta — Pseudoterminal driver. The default is 16. Specify *num* if more than 16 pseudoterminal drivers are defined in the configuation file. For example, to assign 32 pseudoterminals, specify `pseudo-device lta 32` in increments of 16. You must include the `lat`.

decnet — DECNET support – this is required only when the DECNET layered product is installed.

sys_trace — Support of the system call trace capability.

dli — DLI support of `mop_mom` activity.

bsc — Support of 2780/3780 emulation. To work, the `dpv0` or `dup0` devices must be defined in the configuration file as described in Section 1.2.5. (These devices apply to VAX systems only.)

rpc — Remote Procedure Call facility.

nfs — Network File System (NFS) protocol support.

ufs — Local file system support.

scsnet — Systems Communications Services (SCS) network interface driver. For more information, see `scs`(4) in the *ULTRIX Reference Pages*.

audit — This is required when specifying `AUDIT` support. Provides the generation of the `'hostname'/audit.h`, file which causes the appropriate files to be rebuilt when a new system is generated.

sys_tpath — This is required when specifying `SYS_TPATH` support. Provides support for trusted path mechanism.

1.3 Generic Configuration Files

The following examples show typical generic configuration files. Example 1-1 illustrates a VAX configuration. Example 1-2 illustrates a RISC configuration. The

generic configuration file supplied with your system may differ from the one shown here.

Example 1-1: Configuration File for VAX Processors

```
#
# @(#)GENERIC  3.2   (ULTRIX)       6/6/90
# GENERIC VAX
#
machine         vax
cpu       "VAX8800"
cpu       "VAX8600"
cpu       "VAX8200"
cpu       "VAX6400"
cpu       "VAX6200"
cpu       "VAX785"
cpu       "VAX780"
cpu       "VAX750"
cpu       "VAX3600"
cpu       "VAX420"
cpu       "VAX60"
cpu       "MVAX"
ident     GENERIC
timezone        5 dst
maxusers        2
maxuprc           10
physmem         6
processors      1
scs_sysid       32
options         QUOTA
options         INET
options         UFS
options         NFS
options         RPC
options         EMULFLT
options         SCA_SEVERITY   ="SCA_LEVEL5"

config          vmunix          swap on boot
config          dlvmunix  root on boot

#all the adapters and adapter-like items
adapter         xmi0      at   nexus?
adapter         vaxbi0    at   nexus?
adapter         vaxbi1    at   nexus?
adapter         vaxbi2    at   nexus?
adapter         vaxbi3    at   nexus?
adapter         vaxbi4    at   nexus?
adapter         vaxbi5    at   nexus?
adapter         vaxbi11   at   nexus?
adapter         vaxbi12   at   nexus?
adapter         vaxbi13   at   nexus?
adapter         vaxbi14   at   nexus?
adapter         mba0      at   nexus?
adapter         mba1      at   nexus?
adapter         mba2      at   nexus?
adapter         mba3      at   nexus?
adapter         uba0      at   nexus?
adapter         uba1      at   nexus?
adapter         uba2      at   nexus?
adapter         uba3      at   nexus?
adapter         uba4      at   nexus?
adapter         uba5      at   nexus?
adapter         uba6      at   nexus?
```

Example 1-1: (continued)

```
adapter         ibus0    at   nexus?
adapter         ibus1    at   nexus?
adapter         ibus2    at   nexus?
adapter         ibus3    at   nexus?
adapter         ibus4    at   nexus?
adapter         ibus5    at   nexus?
adapter         ibus7    at   nexus?
adapter         msi0     at   nexus?
adapter         ci0      at   nexus?
adapter         ci0      at   vaxbi?     node?

#all the controllers and controller-like items
controller      hsc0       at ci0                  cinode 0
controller      hsc1       at ci0                  cinode 1
controller      hsc2       at ci0                  cinode 2
controller      hsc3       at ci0                  cinode 3
controller      hsc4       at ci0                  cinode 4
controller      hsc5       at ci0                  cinode 5
controller      hsc6       at ci0                  cinode 6
controller      hsc7       at ci0                  cinode 7
controller      hsc8       at ci0                  cinode 8
controller      hsc9       at ci0                  cinode 9
controller      hsc10      at ci0                  cinode 10
controller      hsc11      at ci0                  cinode 11
controller      hsc12      at ci0                  cinode 12
controller      hsc13      at ci0                  cinode 13
controller      hsc14      at ci0                  cinode 14
controller      hsc15      at ci0                  cinode 15
controller      dssc0      at msi0                 msinode 0
controller      dssc1      at msi0                 msinode 1
controller      dssc2      at msi0                 msinode 2
controller      dssc3      at msi0                 msinode 3
controller      dssc4      at msi0                 msinode 4
controller      dssc5      at msi0                 msinode 5
controller      dssc6      at msi0                 msinode 6
controller      dssc7      at msi0                 msinode 7
controller      aio0       at vaxbi?    node?
controller      aio1       at vaxbi?    node?
controller      aie0       at vaxbi?    node?
controller      aie1       at vaxbi?    node?
controller      aie2       at vaxbi?    node?
controller      aie3       at vaxbi?    node?
controller      aie4       at vaxbi?    node?
controller      kdb0       at vaxbi?    node?
controller      kdb1       at vaxbi?    node?
controller      kdb2       at vaxbi?    node?
controller      kdb3       at vaxbi?    node?
controller      kdb4       at vaxbi?    node?
controller      kdb5       at vaxbi?    node?
controller      kdb6       at vaxbi?    node?
controller      kdb7       at vaxbi?    node?
controller      kdb8       at vaxbi?    node?
controller      kdb9       at vaxbi?    node?
controller      kdb10      at vaxbi?    node?
controller      kdb11      at vaxbi?    node?
controller      kdm0       at xmi?      node?
controller      kdm1       at xmi?      node?
controller      klesib0    at vaxbi?    node?
controller      klesib1    at vaxbi?    node?
controller      klesib2    at vaxbi?    node?
controller      klesib3    at vaxbi?    node?
controller      uda0       at uba?
controller      uda1       at uba?
```

Example 1-1: (continued)

```
controller      uda2      at uba?
controller      uda3      at uba?
controller      klesiu0   at uba?
controller      klesiu1   at uba?
controller      klesiu2   at uba?
controller      klesiu3   at uba?
controller      bvpssp0   at aio0       vector bvpsspintr
controller      bvpssp1   at aio1       vector bvpsspintr
controller      bvpssp2   at aie0       vector bvpsspintr
controller      bvpssp3   at aie1       vector bvpsspintr
controller      uq0       at uda0       csr 0172150 vector uqintr
controller      uq1       at uda1       csr 0172150 vector uqintr
controller      uq2       at uda2       csr 0172150 vector uqintr
controller      uq3       at uda3       csr 0172150 vector uqintr
controller      uq4       at kdb0       vector uqintr
controller      uq5       at kdb1       vector uqintr
controller      uq6       at kdb2       vector uqintr
controller      uq7       at kdb3       vector uqintr
controller      uq8       at kdb4       vector uqintr
controller      uq9       at kdb5       vector uqintr
controller      uq10      at kdb6       vector uqintr
controller      uq11      at kdb7       vector uqintr
controller      uq12      at kdb8       vector uqintr
controller      uq13      at kdb9       vector uqintr
controller      uq14      at kdb10      vector uqintr
controller      uq15      at kdb11      vector uqintr
controller      uq16      at klesiu0    csr 0174500 vector uqintr
controller      uq17      at klesiu1    csr 0174500 vector uqintr
controller      uq18      at klesiu2    csr 0174500 vector uqintr
controller      uq19      at klesiu3    csr 0174500 vector uqintr
controller      uq20      at klesib0    vector uqintr
controller      uq21      at klesib1    vector uqintr
controller      uq22      at klesib2    vector uqintr
controller      uq23      at klesib3    vector uqintr
controller      uq24      at kdm0       vector uqintr
controller      uq25      at kdm1       vector uqintr
controller      hk0       at uba?       csr 0177440 vector rkintr
controller      sdc0      at uba0       csr 0x200c0000 vector sdintr
controller      hl0       at uba?       csr 0174400 vector rlintr
controller      zs0       at uba?       csr 0172520 vector tsintr
controller      stc0      at uba0       csr 0x200c0080  vector stintr
controller      scsi0     at uba0       csr 0x200c0080  vector szintr
controller      scsi1     at uba0       csr 0x200c0180  vector szintr
controller      sii0      at ibus?      vector sii_intr

#all the disks
disk      rd0       at sdc0         drive 0
disk      rd1       at sdc0         drive 1
disk      rx2       at sdc0         drive 2
disk      rl0       at hl0          drive 0
disk      rl1       at hl0          drive 1
disk      rl2       at hl0          drive 2
disk      rl3       at hl0          drive 3
disk      hp0       at mba?         drive 0
disk      hp1       at mba?         drive 1
disk      hp2       at mba?         drive 2
disk      hp3       at mba?         drive 3
disk      hp4       at mba?         drive 4
disk      hp5       at mba?         drive 5
disk      hp6       at mba?         drive 6
disk      hp7       at mba?         drive 7
disk      rk0       at hk0          drive 0
disk      rk1       at hk0          drive 1
```

Example 1-1: (continued)

```
disk      rk2      at hk0      drive 2
disk      rk3      at hk0      drive 3
disk      rk4      at hk0      drive 4
disk      rk5      at hk0      drive 5
disk      rk6      at hk0      drive 6
disk      rk7      at hk0      drive 7
disk      ra0      at mscp     drive 0
disk      ra1      at mscp     drive 1
disk      ra2      at mscp     drive 2
disk      ra3      at mscp     drive 3
disk      ra4      at mscp     drive 4
disk      ra5      at mscp     drive 5
disk      ra6      at mscp     drive 6
disk      ra7      at mscp     drive 7
disk      ra8      at mscp     drive 8
disk      ra9      at mscp     drive 9
disk      ra10     at mscp     drive 10
disk      ra11     at mscp     drive 11
disk      ra12     at mscp     drive 12
disk      ra13     at mscp     drive 13
disk      ra14     at mscp     drive 14
disk      ra15     at mscp     drive 15
disk      ra16     at mscp     drive 16
disk      ra17     at mscp     drive 17
disk      ra18     at mscp     drive 18
disk      ra19     at mscp     drive 19
disk      ra20     at mscp     drive 20
disk      ra21     at mscp     drive 21
disk      ra22     at mscp     drive 22
disk      ra23     at mscp     drive 23
disk      ra24     at mscp     drive 24
disk      ra25     at mscp     drive 25
disk      ra26     at mscp     drive 26
disk      ra27     at mscp     drive 27
disk      ra28     at mscp     drive 28
disk      ra29     at mscp     drive 29
disk      ra30     at mscp     drive 30
disk      ra31     at mscp     drive 31
disk      ra32     at mscp     drive 32
disk      ra33     at mscp     drive 33
disk      ra34     at mscp     drive 34
disk      ra35     at mscp     drive 35
disk      ra36     at mscp     drive 36
disk      ra37     at mscp     drive 37
disk      ra38     at mscp     drive 38
disk      ra39     at mscp     drive 39
disk      ra40     at mscp     drive 40
disk      ra41     at mscp     drive 41
disk      ra42     at mscp     drive 42
disk      ra43     at mscp     drive 43
disk      ra44     at mscp     drive 44
disk      ra45     at mscp     drive 45
disk      ra46     at mscp     drive 46
disk      ra47     at mscp     drive 47
disk      ra48     at mscp     drive 48
disk      ra49     at mscp     drive 49
disk      ra50     at mscp     drive 50
disk      ra51     at mscp     drive 51
disk      ra52     at mscp     drive 52
disk      ra53     at mscp     drive 53
disk      ra54     at mscp     drive 54
disk      ra55     at mscp     drive 55
disk      ra56     at mscp     drive 56
```

Example 1-1: (continued)

```
disk      ra57     at mscp      drive 57
disk      ra58     at mscp      drive 58
disk      ra59     at mscp      drive 59
disk      ra60     at mscp      drive 60
disk      ra61     at mscp      drive 61
disk      ra62     at mscp      drive 62
disk      ra63     at mscp      drive 63
disk      ra64     at mscp      drive 64
disk      ra65     at mscp      drive 65
disk      ra66     at mscp      drive 66
disk      ra67     at mscp      drive 67
disk      ra68     at mscp      drive 68
disk      ra69     at mscp      drive 69
disk      ra70     at mscp      drive 70
disk      ra71     at mscp      drive 71
disk      ra72     at mscp      drive 72
disk      ra73     at mscp      drive 73
disk      ra74     at mscp      drive 74
disk      ra75     at mscp      drive 75
disk      ra76     at mscp      drive 76
disk      ra77     at mscp      drive 77
disk      ra78     at mscp      drive 78
disk      ra79     at mscp      drive 79
disk      ra80     at mscp      drive 80
disk      ra81     at mscp      drive 81
disk      ra82     at mscp      drive 82
disk      ra83     at mscp      drive 83
disk      ra84     at mscp      drive 84
disk      ra85     at mscp      drive 85
disk      ra86     at mscp      drive 86
disk      ra87     at mscp      drive 87
disk      ra88     at mscp      drive 88
disk      ra89     at mscp      drive 89
disk      ra90     at mscp      drive 90
disk      ra91     at mscp      drive 91
disk      ra92     at mscp      drive 92
disk      ra93     at mscp      drive 93
disk      ra94     at mscp      drive 94
disk      ra95     at mscp      drive 95
disk      ra96     at mscp      drive 96
disk      ra97     at mscp      drive 97
disk      ra98     at mscp      drive 98
disk      ra99     at mscp      drive 99
disk      ra100    at mscp      drive 100
disk      ra101    at mscp      drive 101
disk      ra102    at mscp      drive 102
disk      ra103    at mscp      drive 103
disk      ra104    at mscp      drive 104
disk      ra105    at mscp      drive 105
disk      ra106    at mscp      drive 106
disk      ra107    at mscp      drive 107
disk      ra108    at mscp      drive 108
disk      ra109    at mscp      drive 109
disk      ra110    at mscp      drive 110
disk      ra111    at mscp      drive 111
disk      ra112    at mscp      drive 112
disk      ra113    at mscp      drive 113
disk      ra114    at mscp      drive 114
disk      ra115    at mscp      drive 115
disk      ra116    at mscp      drive 116
disk      ra117    at mscp      drive 117
disk      ra118    at mscp      drive 118
disk      ra119    at mscp      drive 119
```

Example 1-1: (continued)

```
disk         ra120      at mscp          drive 120
disk         ra121      at mscp          drive 121
disk         ra122      at mscp          drive 122
disk         ra123      at mscp          drive 123
disk         ra124      at mscp          drive 124
disk         ra125      at mscp          drive 125
disk         ra126      at mscp          drive 126
disk         ra127      at mscp          drive 127
disk         ra128      at mscp          drive 128
disk         ra129      at mscp          drive 129
disk         ra130      at mscp          drive 130
disk         ra131      at mscp          drive 131
disk         ra132      at mscp          drive 132
disk         ra133      at mscp          drive 133
disk         ra134      at mscp          drive 134
disk         ra135      at mscp          drive 135
disk         ra136      at mscp          drive 136
disk         ra137      at mscp          drive 137
disk         ra138      at mscp          drive 138
disk         ra139      at mscp          drive 139
disk         ra140      at mscp          drive 140
disk         ra141      at mscp          drive 141
disk         ra142      at mscp          drive 142
disk         ra143      at mscp          drive 143
disk         ra144      at mscp          drive 144
disk         ra145      at mscp          drive 145
disk         ra146      at mscp          drive 146
disk         ra147      at mscp          drive 147
disk         ra148      at mscp          drive 148
disk         ra149      at mscp          drive 149
disk         ra150      at mscp          drive 150
disk         ra151      at mscp          drive 151
disk         ra152      at mscp          drive 152
disk         ra153      at mscp          drive 153
disk         ra154      at mscp          drive 154
disk         ra155      at mscp          drive 155
disk         ra156      at mscp          drive 156
disk         ra157      at mscp          drive 157
disk         ra158      at mscp          drive 158
disk         ra159      at mscp          drive 159
disk         ra160      at mscp          drive 160
disk         ra161      at mscp          drive 161
disk         ra162      at mscp          drive 162
disk         ra163      at mscp          drive 163
disk         ra164      at mscp          drive 164
disk         ra165      at mscp          drive 165
disk         ra166      at mscp          drive 166
disk         ra167      at mscp          drive 167
disk         ra168      at mscp          drive 168
disk         ra169      at mscp          drive 169
disk         ra170      at mscp          drive 170
disk         ra171      at mscp          drive 171
disk         ra172      at mscp          drive 172
disk         ra173      at mscp          drive 173
disk         ra174      at mscp          drive 174
disk         ra175      at mscp          drive 175
disk         ra176      at mscp          drive 176
disk         ra177      at mscp          drive 177
disk         ra178      at mscp          drive 178
disk         ra179      at mscp          drive 179
disk         ra180      at mscp          drive 180
disk         ra181      at mscp          drive 181
disk         ra182      at mscp          drive 182
```

Example 1-1: (continued)

```
disk    ra183   at mscp     drive 183
disk    ra184   at mscp     drive 184
disk    ra185   at mscp     drive 185
disk    ra186   at mscp     drive 186
disk    ra187   at mscp     drive 187
disk    ra188   at mscp     drive 188
disk    ra189   at mscp     drive 189
disk    ra190   at mscp     drive 190
disk    ra191   at mscp     drive 191
disk    ra192   at mscp     drive 192
disk    ra193   at mscp     drive 193
disk    ra194   at mscp     drive 194
disk    ra195   at mscp     drive 195
disk    ra196   at mscp     drive 196
disk    ra197   at mscp     drive 197
disk    ra198   at mscp     drive 198
disk    ra199   at mscp     drive 199
disk    ra200   at mscp     drive 200
disk    ra201   at mscp     drive 201
disk    ra202   at mscp     drive 202
disk    ra203   at mscp     drive 203
disk    ra204   at mscp     drive 204
disk    ra205   at mscp     drive 205
disk    ra206   at mscp     drive 206
disk    ra207   at mscp     drive 207
disk    ra208   at mscp     drive 208
disk    ra209   at mscp     drive 209
disk    ra210   at mscp     drive 210
disk    ra211   at mscp     drive 211
disk    ra212   at mscp     drive 212
disk    ra213   at mscp     drive 213
disk    ra214   at mscp     drive 214
disk    ra215   at mscp     drive 215
disk    ra216   at mscp     drive 216
disk    ra217   at mscp     drive 217
disk    ra218   at mscp     drive 218
disk    ra219   at mscp     drive 219
disk    ra220   at mscp     drive 220
disk    ra221   at mscp     drive 221
disk    ra222   at mscp     drive 222
disk    ra223   at mscp     drive 223
disk    ra224   at mscp     drive 224
disk    ra225   at mscp     drive 225
disk    ra226   at mscp     drive 226
disk    ra227   at mscp     drive 227
disk    ra228   at mscp     drive 228
disk    ra229   at mscp     drive 229
disk    ra230   at mscp     drive 230
disk    ra231   at mscp     drive 231
disk    ra232   at mscp     drive 232
disk    ra233   at mscp     drive 233
disk    ra234   at mscp     drive 234
disk    ra235   at mscp     drive 235
disk    ra236   at mscp     drive 236
disk    ra237   at mscp     drive 237
disk    ra238   at mscp     drive 238
disk    ra239   at mscp     drive 239
disk    ra240   at mscp     drive 240
disk    ra241   at mscp     drive 241
disk    ra242   at mscp     drive 242
disk    ra243   at mscp     drive 243
disk    ra244   at mscp     drive 244
disk    ra245   at mscp     drive 245
```

Example 1-1: (continued)

```
disk        ra246      at mscp          drive 246
disk        ra247      at mscp          drive 247
disk        ra248      at mscp          drive 248
disk        ra249      at mscp          drive 249
disk        ra250      at mscp          drive 250
disk        ra251      at mscp          drive 251
disk        ra252      at mscp          drive 252
disk        ra253      at mscp          drive 253
disk        ra254      at mscp          drive 254
disk        rz0        at scsi0    drive 0
disk        rz1        at scsi0    drive 1
disk        rz2        at scsi0    drive 2
disk        rz3        at scsi0    drive 3
disk        rz4        at scsi0    drive 4
disk        rz5        at scsi0    drive 5
disk        rz6        at scsi0    drive 6
disk        rz7        at scsi0    drive 7
disk        rz8        at scsi1    drive 0
disk        rz9        at scsi1    drive 1
disk        rz10       at scsi1    drive 2
disk        rz11       at scsi1    drive 3
disk        rz12       at scsi1    drive 4
disk        rz13       at scsi1    drive 5
disk        rz14       at scsi1    drive 6
disk        rz15       at scsi1    drive 7
disk        rz0        at sii0          drive 0
disk        rz1        at sii0          drive 1
disk        rz2        at sii0          drive 2
disk        rz3        at sii0          drive 3
disk        rz4        at sii0          drive 4
disk        rz5        at sii0          drive 5
disk        rz6        at sii0          drive 6
disk        rz7        at sii0          drive 7

#all the tapes
tape        st0        at stc0          drive 0
tape        ts0        at zs0           drive 0
master           ht0        at mba?          drive ?
tape        tu0        at ht0           slave 0
tape        tu1        at ht0           slave 1
tape        tu2        at ht0           slave 2
tape        tu3        at ht0           slave 3
master            mt0        at mba?         drive ?
tape        mu0        at mt0           slave 0
tape        mu1        at mt0           slave 1
tape        mu2        at mt0           slave 2
tape        mu3        at mt0           slave 3
tape        tms0       at mscp          drive 0
tape        tms1       at mscp          drive 1
tape        tms2       at mscp          drive 2
tape        tms3       at mscp          drive 3
tape        tms4       at mscp          drive 4
tape        tms5       at mscp          drive 5
tape        tms6       at mscp          drive 6
tape        tms7       at mscp          drive 7
tape        tms8       at mscp          drive 8
tape        tms9       at mscp          drive 9
tape        tms10      at mscp          drive 10
tape        tms11      at mscp          drive 11
tape        tms12      at mscp          drive 12
tape        tms13      at mscp          drive 13
tape        tms14      at mscp          drive 14
tape        tms15      at mscp          drive 15
```

Example 1-1: (continued)

```
tape      tms16     at mscp        drive 16
tape      tms17     at mscp        drive 17
tape      tms18     at mscp        drive 18
tape      tms19     at mscp        drive 19
tape      tms20     at mscp        drive 20
tape      tms21     at mscp        drive 21
tape      tms22     at mscp        drive 22
tape      tms23     at mscp        drive 23
tape      tms24     at mscp        drive 24
tape      tms25     at mscp        drive 25
tape      tms26     at mscp        drive 26
tape      tms27     at mscp        drive 27
tape      tms28     at mscp        drive 28
tape      tms29     at mscp        drive 29
tape      tms30     at mscp        drive 30
tape      tms31     at mscp        drive 31
tape      tz0       at scsi0             drive 0
tape      tz1       at scsi0             drive 1
tape      tz2       at scsi0             drive 2
tape      tz3       at scsi0             drive 3
tape      tz4       at scsi0             drive 4
tape      tz5       at scsi0             drive 5
tape      tz6       at scsi0             drive 6
tape      tz7       at scsi0             drive 7
tape      tz8       at scsi1             drive 0
tape      tz9       at scsi1             drive 1
tape      tz10      at scsi1             drive 2
tape      tz11      at scsi1             drive 3
tape      tz12      at scsi1             drive 4
tape      tz13      at scsi1             drive 5
tape      tz14      at scsi1             drive 6
tape      tz15      at scsi1             drive 7

#all the workstations
device  qv0 at uba0  csr 0177200 flags 0x0f vector qvkint qvvint
device  qd0 at uba0  csr 0177400 flags 0x0f vector qddint qdaint qdiint
device  qd1 at uba0  csr 0177402 flags 0x0f vector qddint qdaint qdiint
device  sm0 at uba0  csr 0x200f0000 flags 0x0f vector smvint
device  sg0 at uba0  csr 0x3c000000 flags 0x0f vector sgaint sgfint
device  fg0 at ibus? flags 0x0f vector fgvint

#all the networks
device       bvpni0    at aie0        vector bvpniintr
device       bvpni1    at aie2        vector bvpniintr
device       bvpni2    at aie3        vector bvpniintr
device       bvpni3    at aie4        vector bvpniintr
device       xna0 at vaxbi? node? vector xnaintr
device       xna1 at vaxbi? node? vector xnaintr
device       xna2 at vaxbi? node? vector xnaintr
device       xna3 at vaxbi? node? vector xnaintr
device       xna4 at xmi?        node? vector xnaintr
device       xna5 at xmi?        node? vector xnaintr
device       xna6 at xmi?        node? vector xnaintr
device       xna7 at xmi?        node? vector xnaintr
device       de0  at uba?        csr 0174510 vector deintr
device       de1  at uba?        csr 0174510 vector deintr
device       qe0  at uba0        csr 0174440 vector qeintr
device       qe1  at uba0        csr 0174460 vector qeintr
device       ln0  at ibus?  vector lnintr

#all the terminals and printers
device  fc0   at ibus?  flags 0x0f vector fcxrint
device  ss0   at uba?   csr 0x200a0000 flags 0x0f vector ssrint ssxint
```

Example 1-1: (continued)

```
device  sh0   at uba0   csr 0x38000000 flags 0xff vector shrint shxint
device  lp0   at uba?   csr 0177514 vector lpintr
device  dmb0  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb1  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb2  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb3  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb4  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb5  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb6  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb7  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb8  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb9  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb10 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb11 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb12 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb13 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb14 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb15 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint

#all the pseudo items
pseudo-device   pty
pseudo-device   loop
pseudo-device   inet
pseudo-device   ether
pseudo-device   ufs
pseudo-device   nfs
pseudo-device   rpc
pseudo-device   presto
```

Example 1-2: Configuration File for RISC Processors

```
#
# @(#)GENERIC   3.6   (ULTRIX)        6/15/90
# GENERIC RISC
#
machine         mips
cpu         "DS3100"
cpu         "DS5400"
cpu         "DS5500"
cpu         "DS5800"
cpu         "DS5000"
ident       "GENERIC"
timezone         5 dst
maxusers         32
maxuprc          50
physmem          8
processors       1
scs_sysid        1

options         QUOTA
options         INET
options         NFS
options         UFS
options         RPC
options         SYS_TRACE
options         LAT
options         DLI
options         UWS

makeoptions     ENDIAN="-EL"
```

Example 1-2: (continued)

```
config          vmunix          swap on boot
config          dlgenvmunix     root on ln0
config          dlsgenvmunix    root on ln0 swap on rz0b
config          rzzerovmunix    root on rz0a swap on rz0b

#all the adapters and adapter-like items
adapter         xmi0    at nexus?
adapter         vaxbi0  at nexus?
adapter         vaxbi1  at nexus?
adapter         vaxbi2  at nexus?
adapter         vaxbi3  at nexus?
adapter         vaxbi4  at nexus?
adapter         vaxbi5  at nexus?
adapter         vaxbi11 at nexus?
adapter         vaxbi12 at nexus?
adapter         vaxbi13 at nexus?
adapter         vaxbi14 at nexus?
adapter         uba0    at nexus?
adapter         uba1    at nexus?
adapter         uba2    at nexus?
adapter         uba3    at nexus?
adapter         uba4    at nexus?
adapter         uba5    at nexus?
adapter         uba6    at nexus?
adapter         ibus0   at nexus?
adapter         ibus1   at nexus?
adapter         ibus2   at nexus?
adapter         ibus3   at nexus?
adapter         ibus4   at nexus?
adapter         ibus5   at nexus?
adapter         ibus6   at nexus?
adapter         ibus7   at nexus?
adapter         msi0    at nexus?
adapter         ci0     at nexus?
adapter         vba0    at nexus?
adapter         vba1    at nexus?
adapter         vba2    at nexus?
adapter         vba3    at nexus?

#all the controllers and controller-like items
controller  hsc0    at ci0          cinode 0
controller  hsc1    at ci0          cinode 1
controller  hsc2    at ci0          cinode 2
controller  hsc3    at ci0          cinode 3
controller  hsc4    at ci0          cinode 4
controller  hsc5    at ci0          cinode 5
controller  hsc6    at ci0          cinode 6
controller  hsc7    at ci0          cinode 7
controller  hsc8    at ci0          cinode 8
controller  hsc9    at ci0          cinode 9
controller  hsc10   at ci0          cinode 10
controller  hsc11   at ci0          cinode 11
controller  hsc12   at ci0          cinode 12
controller  hsc13   at ci0          cinode 13
controller  hsc14   at ci0          cinode 14
controller  hsc15   at ci0          cinode 15
controller  aio0    at vaxbi? node?
controller  aio1    at vaxbi? node?
controller  aie0    at vaxbi? node?
controller  aie1    at vaxbi? node?
controller  aie2    at vaxbi? node?
controller  aie3    at vaxbi? node?
controller  aie4    at vaxbi? node?
```

Example 1-2: (continued)

```
controller   kdb0     at vaxbi? node?
controller   kdb1     at vaxbi? node?
controller   kdb2     at vaxbi? node?
controller   kdb3     at vaxbi? node?
controller   kdb4     at vaxbi? node?
controller   kdb5     at vaxbi? node?
controller   kdb6     at vaxbi? node?
controller   kdb7     at vaxbi? node?
controller   kdb8     at vaxbi? node?
controller   kdb9     at vaxbi? node?
controller         kdb10   at vaxbi? node?
controller   kdb11    at vaxbi? node?
controller   klesib0 at vaxbi? node?
controller   klesib1 at vaxbi? node?
controller   klesib2 at vaxbi? node?
controller   klesib3 at vaxbi? node?
controller        kdm0    at xmi?        node?
controller   kdm1     at xmi?        node?
controller   kdm2     at xmi?        node?
controller   kdm3     at xmi?        node?
controller   uda0     at uba?
controller   uda1     at uba?
controller   uda2     at uba?
controller   uda3     at uba?
controller   klesiu0 at uba?
controller   klesiu1 at uba?
controller   klesiu2 at uba?
controller   klesiu3 at uba?
controller   bvpssp0 at aio0        vector bvpsspintr
controller   bvpssp1 at aio1        vector bvpsspintr
controller   bvpssp2 at aie0        vector bvpsspintr
controller   bvpssp3 at aie1        vector bvpsspintr
controller   uq0at uda0        csr 0172150 vector uqintr
controller   uq1at uda1        csr 0172150 vector uqintr
controller   uq2at uda2        csr 0172150 vector uqintr
controller   uq3at uda3        csr 0172150 vector uqintr
controller   uq4at kdb0        vector uqintr
controller   uq5at kdb1        vector uqintr
controller   uq6at kdb2        vector uqintr
controller   uq7at kdb3        vector uqintr
controller   uq8at kdb4        vector uqintr
controller   uq9at kdb5        vector uqintr
controller   uq10     at kdb6        vector uqintr
controller   uq11     at kdb7        vector uqintr
controller   uq12     at kdb8        vector uqintr
controller   uq13     at kdb9        vector uqintr
controller   uq14     at kdb10  vector uqintr
controller   uq15     at kdb11  vector uqintr
controller   uq16     at klesiu0      csr 0174500 vector uqintr
controller   uq17     at klesiu1      csr 0174500 vector uqintr
controller   uq18     at klesiu2      csr 0174500 vector uqintr
controller   uq19     at klesiu3      csr 0174500 vector uqintr
controller   uq20     at klesib0      vector uqintr
controller   uq21     at klesib1      vector uqintr
controller   uq22     at klesib2      vector uqintr
controller   uq23     at klesib3      vector uqintr
controller   uq24     at kdm0         vector uqintr
controller   uq25     at kdm1         vector uqintr
controller   uq26     at kdm2         vector uqintr
controller   uq27     at kdm3         vector uqintr
controller   dssc0    at msi0        msinode 0
controller   dssc1    at msi0        msinode 1
controller   dssc2    at msi0        msinode 2
```

Example 1-2: (continued)

```
controller  dssc3   at msi0       msinode 3
controller  dssc4   at msi0       msinode 4
controller  dssc5   at msi0       msinode 5
controller  dssc6   at msi0       msinode 6
controller  dssc7   at msi0       msinode 7
controller  uda0    at uba?
controller  uda1    at uba?
controller  uda2    at uba?
controller  uda3    at uba?
controller  klesiu0 at uba?
controller  klesiu1 at uba?
controller  klesiu2 at uba?
controller  klesiu3 at uba?
controller  sii0    at ibus?  vector sii_intr
controller  asc0    at ibus?  vector ascintr
controller      asc1    at ibus?  vector ascintr
controller      asc2    at ibus?  vector ascintr
controller  asc3    at ibus?  vector ascintr

disk      ra0       at mscp       drive 0
disk      ra1    at mscp      drive 1
disk      ra2    at mscp      drive 2
disk      ra3    at mscp      drive 3
disk      ra4    at mscp      drive 4
disk      ra5    at mscp      drive 5
disk      ra6    at mscp      drive 6
disk      ra7    at mscp      drive 7
disk      ra8    at mscp      drive 8
disk      ra9    at mscp      drive 9
disk      ra10   at mscp      drive 10
disk      ra11   at mscp      drive 11
disk      ra12   at mscp      drive 12
disk      ra13   at mscp      drive 13
disk      ra14   at mscp      drive 14
disk      ra15   at mscp      drive 15
disk      ra16   at mscp      drive 16
disk      ra17   at mscp      drive 17
disk      ra18   at mscp      drive 18
disk      ra19   at mscp      drive 19
disk      ra20   at mscp      drive 20
disk      ra21   at mscp      drive 21
disk      ra22   at mscp      drive 22
disk      ra23   at mscp      drive 23
disk      ra24   at mscp      drive 24
disk      ra25   at mscp      drive 25
disk      ra26   at mscp      drive 26
disk      ra27   at mscp      drive 27
disk      ra28   at mscp      drive 28
disk      ra29   at mscp      drive 29
disk      ra30   at mscp      drive 30
disk      ra31   at mscp      drive 31
disk      ra32   at mscp      drive 32
disk      ra33        at mscp      drive 33
disk      ra34      at mscp       drive 34
disk      ra35   at mscp      drive 35
disk      ra36   at mscp      drive 36
disk      ra37   at mscp      drive 37
disk      ra38   at mscp      drive 38
disk      ra39   at mscp      drive 39
disk      ra40   at mscp      drive 40
disk      ra41   at mscp      drive 41
disk      ra42   at mscp      drive 42
disk      ra43   at mscp      drive 43
```

Example 1-2: (continued)

```
disk      ra44    at mscp      drive 44
disk      ra45    at mscp      drive 45
disk      ra46    at mscp      drive 46
disk      ra47    at mscp      drive 47
disk      ra48    at mscp      drive 48
disk      ra49    at mscp      drive 49
disk      ra50    at mscp      drive 50
disk      ra51    at mscp      drive 51
disk      ra52    at mscp      drive 52
disk      ra53    at mscp      drive 53
disk      ra54    at mscp      drive 54
disk      ra55    at mscp      drive 55
disk      ra56    at mscp      drive 56
disk      ra57    at mscp      drive 57
disk      ra58    at mscp      drive 58
disk      ra59    at mscp      drive 59
disk      ra60    at mscp      drive 60
disk      ra61    at mscp      drive 61
disk      ra62    at mscp      drive 62
disk      ra63    at mscp      drive 63
disk      ra64    at mscp      drive 64
disk      ra65    at mscp      drive 65
disk      ra66    at mscp      drive 66
disk      ra67    at mscp      drive 67
disk      ra68    at mscp      drive 68
disk      ra69    at mscp      drive 69
disk      ra70    at mscp      drive 70
disk      ra71    at mscp      drive 71
disk      ra72    at mscp      drive 72
disk      ra73    at mscp      drive 73
disk      ra74    at mscp      drive 74
disk      ra75    at mscp      drive 75
disk      ra76    at mscp      drive 76
disk      ra77    at mscp      drive 77
disk      ra78    at mscp      drive 78
disk      ra79    at mscp      drive 79
disk      ra80    at mscp      drive 80
disk      ra81    at mscp      drive 81
disk      ra82    at mscp      drive 82
disk      ra83    at mscp      drive 83
disk      ra84    at mscp      drive 84
disk      ra85    at mscp      drive 85
disk      ra86    at mscp      drive 86
disk      ra87    at mscp      drive 87
disk      ra88    at mscp      drive 88
disk      ra89    at mscp      drive 89
disk      ra90    at mscp      drive 90
disk      ra91    at mscp      drive 91
disk      ra92    at mscp      drive 92
disk      ra93    at mscp      drive 93
disk      ra94    at mscp      drive 94
disk      ra95    at mscp      drive 95
disk      ra96    at mscp      drive 96
disk      ra97    at mscp      drive 97
disk      ra98    at mscp      drive 98
disk      ra99    at mscp      drive 99
disk      ra100   at mscp      drive 100
disk      ra101   at mscp      drive 101
disk      ra102   at mscp      drive 102
disk      ra103   at mscp      drive 103
disk      ra104   at mscp      drive 104
disk      ra105   at mscp      drive 105
disk      ra106   at mscp      drive 106
```

Example 1-2: (continued)

```
disk        ra107  at mscp      drive 107
disk        ra108  at mscp      drive 108
disk        ra109  at mscp      drive 109
disk        ra110  at mscp      drive 110
disk        ra111  at mscp      drive 111
disk        ra112  at mscp      drive 112
disk        ra113  at mscp      drive 113
disk        ra114  at mscp      drive 114
disk        ra115  at mscp      drive 115
disk        ra116  at mscp      drive 116
disk        ra117  at mscp      drive 117
disk        ra118  at mscp      drive 118
disk        ra119  at mscp      drive 119
disk        ra120  at mscp      drive 120
disk        ra121  at mscp      drive 121
disk        ra122  at mscp      drive 122
disk        ra123  at mscp      drive 123
disk        ra124  at mscp      drive 124
disk        ra125  at mscp      drive 125
disk        ra126  at mscp      drive 126
disk        ra127      at mscp      drive 127
disk        ra128  at mscp      drive 128
disk        ra129  at mscp      drive 129
disk        ra130  at mscp      drive 130
disk        ra131  at mscp      drive 131
disk        ra132  at mscp      drive 132
disk        ra133  at mscp      drive 133
disk        ra134  at mscp      drive 134
disk        ra135  at mscp      drive 135
disk        ra136  at mscp      drive 136
disk        ra137  at mscp      drive 137
disk        ra138  at mscp      drive 138
disk        ra139  at mscp      drive 139
disk        ra140  at mscp      drive 140
disk        ra141  at mscp      drive 141
disk        ra142  at mscp      drive 142
disk        ra143  at mscp      drive 143
disk        ra144  at mscp      drive 144
disk        ra145  at mscp      drive 145
disk        ra146  at mscp      drive 146
disk        ra147  at mscp      drive 147
disk        ra148  at mscp      drive 148
disk        ra149  at mscp      drive 149
disk        ra150  at mscp      drive 150
disk        ra151  at mscp      drive 151
disk        ra152  at mscp      drive 152
disk        ra153  at mscp      drive 153
disk        ra154  at mscp      drive 154
disk        ra155  at mscp      drive 155
disk        ra156  at mscp      drive 156
disk        ra157  at mscp      drive 157
disk        ra158  at mscp      drive 158
disk        ra159  at mscp      drive 159
disk        ra160  at mscp      drive 160
disk        ra161  at mscp      drive 161
disk        ra162  at mscp      drive 162
disk        ra163  at mscp      drive 163
disk        ra164  at mscp      drive 164
disk        ra165  at mscp      drive 165
disk        ra166  at mscp      drive 166
disk        ra167  at mscp      drive 167
disk        ra168  at mscp      drive 168
disk        ra169  at mscp      drive 169
```

Example 1-2: (continued)

```
disk        ra170  at mscp       drive 170
disk        ra171  at mscp       drive 171
disk        ra172  at mscp       drive 172
disk        ra173  at mscp       drive 173
disk        ra174  at mscp       drive 174
disk        ra175  at mscp       drive 175
disk        ra176  at mscp       drive 176
disk        ra177  at mscp       drive 177
disk        ra178  at mscp       drive 178
disk        ra179  at mscp       drive 179
disk        ra180  at mscp       drive 180
disk        ra181  at mscp       drive 181
disk        ra182  at mscp       drive 182
disk        ra183  at mscp       drive 183
disk        ra184  at mscp       drive 184
disk        ra185  at mscp       drive 185
disk        ra186  at mscp       drive 186
disk        ra187  at mscp       drive 187
disk        ra188  at mscp       drive 188
disk        ra189  at mscp       drive 189
disk        ra190  at mscp       drive 190
disk        ra191  at mscp       drive 191
disk        ra192  at mscp       drive 192
disk        ra193  at mscp       drive 193
disk        ra194  at mscp       drive 194
disk        ra195  at mscp       drive 195
disk        ra196  at mscp       drive 196
disk        ra197  at mscp       drive 197
disk        ra198  at mscp       drive 198
disk        ra199  at mscp       drive 199
disk        ra200  at mscp       drive 200
disk        ra201  at mscp       drive 201
disk        ra202  at mscp       drive 202
disk        ra203  at mscp       drive 203
disk        ra204  at mscp       drive 204
disk        ra205  at mscp       drive 205
disk        ra206  at mscp       drive 206
disk        ra207  at mscp       drive 207
disk        ra208  at mscp       drive 208
disk        ra209  at mscp       drive 209
disk        ra210  at mscp       drive 210
disk        ra211  at mscp       drive 211
disk        ra212  at mscp       drive 212
disk        ra213  at mscp       drive 213
disk        ra214  at mscp       drive 214
disk        ra215  at mscp       drive 215
disk        ra216  at mscp       drive 216
disk        ra217  at mscp       drive 217
disk        ra218  at mscp       drive 218
disk        ra219  at mscp       drive 219
disk        ra220  at mscp       drive 220
disk        ra221  at mscp       drive 221
disk        ra222  at mscp       drive 222
disk        ra223  at mscp       drive 223
disk        ra224  at mscp       drive 224
disk        ra225  at mscp       drive 225
disk        ra226  at mscp       drive 226
disk        ra227  at mscp       drive 227
disk        ra228  at mscp       drive 228
disk        ra229  at mscp       drive 229
disk        ra230  at mscp       drive 230
disk        ra231  at mscp       drive 231
disk        ra232  at mscp       drive 232
```

Example 1-2: (continued)

```
disk        ra233  at mscp       drive 233
disk        ra234  at mscp       drive 234
disk        ra235  at mscp       drive 235
disk        ra236  at mscp       drive 236
disk        ra237  at mscp       drive 237
disk        ra238  at mscp       drive 238
disk        ra239  at mscp       drive 239
disk        ra240  at mscp       drive 240
disk        ra241  at mscp       drive 241
disk        ra242  at mscp       drive 242
disk        ra243  at mscp       drive 243
disk        ra244  at mscp       drive 244
disk        ra245  at mscp       drive 245
disk        ra246  at mscp       drive 246
disk        ra247  at mscp       drive 247
disk        ra248  at mscp       drive 248
disk        ra249  at mscp       drive 249
disk        ra250  at mscp       drive 250
disk        ra251  at mscp       drive 251
disk        ra252  at mscp       drive 252
disk        ra253  at mscp       drive 253
disk        ra254  at mscp       drive 254
disk        rz0        at sii0         drive 0
disk        rz1        at sii0         drive 1
disk        rz2        at sii0         drive 2
disk        rz3        at sii0         drive 3
disk        rz4        at sii0         drive 4
disk        rz5        at sii0         drive 5
disk        rz6        at sii0         drive 6
disk        rz7    at sii0       drive 7
disk        rz0    at asc0       drive 0
disk        rz1    at asc0       drive 1
disk        rz2    at asc0       drive 2
disk        rz3    at asc0       drive 3
disk        rz4    at asc0       drive 4
disk        rz5    at asc0       drive 5
disk        rz6    at asc0       drive 6
disk        rz7    at asc0       drive 7
disk        rz8    at asc1       drive 0
disk        rz9    at asc1       drive 1
disk        rz10   at asc1       drive 2
disk        rz11   at asc1       drive 3
disk        rz12   at asc1       drive 4
disk        rz13   at asc1       drive 5
disk        rz14   at asc1       drive 6
disk        rz15   at asc1       drive 7
disk        rz16   at asc2       drive 0
disk        rz17   at asc2       drive 1
disk        rz18   at asc2       drive 2
disk        rz19   at asc2       drive 3
disk        rz20   at asc2       drive 4
disk        rz21   at asc2       drive 5
disk        rz22   at asc2       drive 6
disk        rz23   at asc2       drive 7
disk        rz24   at asc3       drive 0
disk        rz25   at asc3       drive 1
disk        rz26   at asc3       drive 2
disk        rz27   at asc3       drive 3
disk        rz28   at asc3       drive 4
disk        rz29   at asc3       drive 5
disk        rz30   at asc3       drive 6
disk        rz31   at asc3       drive 7
```

Example 1-2: (continued)

```
tape      tms0 at mscp          drive 0
tape      tms1 at mscp          drive 1
tape      tms2 at mscp          drive 2
tape      tms3 at mscp          drive 3
tape      tms4 at mscp          drive 4
tape      tms5 at mscp          drive 5
tape      tms6 at mscp          drive 6
tape      tms7 at mscp          drive 7
tape      tms8 at mscp          drive 8
tape      tms9 at mscp          drive 9
tape      tms10   at mscp       drive 10
tape      tms11   at mscp       drive 11
tape      tms12   at mscp       drive 12
tape      tms13   at mscp       drive 13
tape      tms14   at mscp       drive 14
tape      tms15   at mscp       drive 15
tape      tms16   at mscp       drive 16
tape      tms17   at mscp       drive 17
tape      tms18   at mscp       drive 18
tape      tms19   at mscp       drive 19
tape      tms20   at mscp       drive 20
tape      tms21   at mscp       drive 21
tape      tms22   at mscp       drive 22
tape      tms23   at mscp       drive 23
tape      tms24   at mscp       drive 24
tape      tms25   at mscp       drive 25
tape      tms26   at mscp       drive 26
tape      tms27   at mscp       drive 27
tape      tms28   at mscp       drive 28
tape      tms29   at mscp       drive 29
tape      tms30   at mscp       drive 30
tape      tms31   at mscp       drive 31
tape      tz0     at sii0           drive 0
tape      tz1     at sii0           drive 1
tape      tz2     at sii0           drive 2
tape      tz3     at sii0           drive 3
tape      tz4     at sii0           drive 4
tape      tz5     at sii0           drive 5
tape      tz6     at sii0           drive 6
tape      tz7     at sii0           drive 7
tape      tz0     at asc0           drive 0
tape      tz1     at asc0           drive 1
tape      tz2     at asc0           drive 2
tape      tz3     at asc0           drive 3
tape      tz4     at asc0           drive 4
tape      tz5     at asc0           drive 5
tape      tz6     at asc0           drive 6
tape      tz7  at asc0          drive 7
tape      tz8     at asc1           drive 0
tape      tz9     at asc1           drive 1
tape      tz10    at asc1           drive 2
tape      tz11    at asc1           drive 3
tape      tz12    at asc1           drive 4
tape      tz13    at asc1           drive 5
tape      tz14    at asc1           drive 6
tape      tz15 at asc1          drive 7
tape      tz16    at asc2           drive 0
tape      tz17    at asc2           drive 1
tape      tz18    at asc2           drive 2
tape      tz19    at asc2           drive 3
tape      tz20    at asc2           drive 4
tape      tz21    at asc2           drive 5
tape      tz22    at asc2           drive 6
```

Example 1-2: (continued)

```
tape      tz23 at asc2        drive 7
tape      tz24    at asc3        drive 0
tape      tz25    at asc3        drive 1
tape      tz26    at asc3        drive 2
tape      tz27    at asc3        drive 3
tape      tz28    at asc3        drive 4
tape      tz29    at asc3        drive 5
tape      tz30    at asc3        drive 6
tape      tz31 at asc3        drive 7

# Ethernet devices
device          xna0 at vaxbi? node? vector xnaintr
device          xna1 at vaxbi? node? vector xnaintr
device          xna2 at vaxbi? node? vector xnaintr
device          xna3 at vaxbi? node? vector xnaintr
device          ln0  at ibus?  vector lnintr
device           ln1     at ibus?        vector lnintr
device           ln2     at ibus?        vector lnintr
device           ln3     at ibus?        vector lnintr
device          ne0  at ibus?  vector neintr
device          fza0 at ibus?  vector fzaintr
device          fza1 at ibus?  vector fzaintr
device          fza2 at ibus?  vector fzaintr
device          qe0  at uba?        csr 0174440 vector qeintr
device          qe1  at uba?        csr 0174460 vector qeintr

# Terminal Devices
device  dmb0  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb1  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb2  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb3  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb4  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb5  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb6  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb7  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb8  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb9  at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb10 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb11 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb12 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb13 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb14 at vaxbi? node? flags 0xff vector dmbsint dmbaint dmblint
device  dmb15 at vaxbi?node? flags 0xff vector dmbsint dmbaint dmblint
device          dc0  at ibus?  vector dcintr
device          mdc0 at ibus?  vector mdcintr
device          mdc1 at ibus?  vector mdcintr
device          mdc2 at ibus?  vector mdcintr

# graphic devices
device          pm0     at ibus?        vector pmvint
device          cfb0    at ibus?        vector cfbvint
device          gq0     at ibus?        vector gqintr
device          ga0     at ibus?        vector gaintr

# Other devices
device          lp0     at uba?         csr 0177514 vector lpintr

pseudo-device  pty  64
pseudo-device  inet
pseudo-device  ether
pseudo-device  loop
pseudo-device  nfs
pseudo-device  ufs
```

Example 1-2: (continued)

```
pseudo-device   rpc
pseudo-device   sys_trace
pseudo-device   lat
pseudo-device   lta   32
pseudo-device   dli
pseudo-device    presto
```

Building the Kernel 2

This chapter describes how to build a kernel. There are three ways to build a kernel:

- You can build a new kernel automatically, using the `doconfig` command. Section 2.2 describes this procedure.
- You can build the kernel manually, following the steps listed in Section 2.3. If you opt to build the kernel manually, make sure that you understand the contents and format of the configuration file. Chapter 1 describes this file.
- You can build a kernel when you change the user capacity of your system, by using either the License Management Facility or the capacity upgrade installation. Section 2.4 describes this procedure.

Choose the procedure that best complements your experience and the needs of your particular installation. Should the new kernel you build fail to boot, you can use the procedure described in Section 2.5 to recover the original kernel.

Note

In this chapter, there are two naming conventions:

- The string `HOSTNAME` represents the name you have assigned to your system, in uppercase letters.
- The string `{vax,mips}` or `{VAX,MIPS}` represents separate directory paths. You choose which directory path to use, depending on your machine's architecture, VAX or RISC.

2.1 When To Build a New Kernel

You need to build a new kernel after any of the following events:

- If you add a new device and its driver to your configuration. When you add a new device and device driver, you need to rebuild the kernel to include the specifications in the configuration file.
- If you delete a device and its driver from your configuration. When you delete a device and device driver from your configuration and edit the configuration file to include only the actual hardware and software at your installation, you need to rebuild the kernel to match this configuration.
- If you tune the operating system. When you alter the default configuration or change the original disk setup, you need to rebuild the kernel. For example, if you create swap areas on two disk drives, thereby modifying the original single swap area on disk, you need to rebuild the kernel.

You may need to build a new kernel after any of these events:

- If you upgrade your system. For example, if you increase the login capacity on your system through the License Management Facility or a capacity upgrade, you may need to rebuild the kernel.
- If you add layered products. For example, if you add the DECnet facility, or any layered product that requires system configuration changes, you may need to rebuild the kernel.

2.2 Building a Kernel Automatically

The ULTRIX software provides the `/etc/doconfig` program with which you build your kernel automatically. The following section describes this procedure.

2.2.1 Using the doconfig Program

To update an existing configuration file or create a new one using `/etc/doconfig`, the system must be operating the generic kernel, `genvmunix`.

To use the `/etc/doconfig` program, follow these steps:

1. Log in as superuser (`root`). You must be superuser to execute the `doconfig` command.

2. Shut the system down to single-user mode by typing:

   ```
   # /etc/shutdown +5  "Building a new kernel"
   ```

3. Save the running `vmunix` as `vmunix.old` by typing:

   ```
   # mv /vmunix /sys/vmunix.old
   ```

4. Copy `/genvmunix` to `/vmunix` by typing:

   ```
   # cp /genvmunix /vmunix
   ```

5. Halt the processor by typing:

   ```
   #  /etc/halt
   ```

6. Reboot the system to single-user mode. Refer to the *Guide to Shutdown and Startup* for instructions (different processors have different procedures).

7. Check the file systems:

   ```
   # /etc/fsck -p
   ```

8. Mount the UFS file systems:

   ```
   # /etc/mount -a -t ufs
   ```

9. Start the error log, by entering:

   ```
   # eli -s
   ```

   The following question is displayed:

   ```
   Caution:  Are you in Single User Mode? (y)
   ```

Answer yes to the question; you shut down to single-user mode in step 2.

10. Run the update daemon by issuing the following command:

```
# /etc/update
```

11. Save your existing configuration file:

```
# cd /sys/conf/{vax,mips}
# cp HOSTNAME HOSTNAME.old
```

12. Set the EDITOR environment variable to specify the text editor you want to use to edit the configuration file. For example, to use the `ex` editor, type:

```
# EDITOR=ex
# export EDITOR
```

13. Run the `doconfig` program by typing:

```
# cd /
# /etc/doconfig
```

The `doconfig` program then prompts you for information about your system configuration.

14. Type yes when the `doconfig` program asks if you want to edit the configuration file. The `doconfig` program then invokes the editor specified by the EDITOR environment variable.

15. Compare your saved configuration file with the new configuration file to determine the differences (if any). You can use the editor's shell escape to compare the files. For example, if you are using the `ex` editor, type:

```
!diff /sys/conf/{vax, mips}/HOSTNAME /sys/conf/{vax, mips}/HOSTNAME.old
```

16. Edit the new configuration file to include the kernel options, pseudo-devices, system parameters, or other changes you want to bring forward from the old configuration file.

Note

If you added or removed any communications devices from your configuration file you need to edit the `/etc/ttys` file to match your new configuration (that is, to match the `/dev/tty??` files).

17. Write the changes to the new configuration file and end the editing session. The `doconfig` program will build the new kernel. When the `doconfig` program finishes, it prints a message showing the path and location of the new `vmunix`. To test the new kernel, see Section 2.2.2.

Refer to `doconfig`(8) in the *ULTRIX Reference Pages* for details on the command and its options.

Example 2-1 shows a sample execution of the `doconfig` program. Entries in square brackets ([ ]) are the default values. To select a default value, press the RETURN key. The example shows the default entries typed in for presentation purposes only.

Once you enter the system name and the date and time information, the `doconfig` program builds a configuration file. Note that if you type a system name that exists, the `doconfig` program will verify that you want that system replaced. If you provide the name of a system that does not exist, you are not asked this question. When `doconfig` completes the configuration file build process, it loads `vmunix`, rearranges the symbol table, and makes the special files for the system based on the configuration.

Example 2-1: Sample doconfig Execution

```
# /etc/doconfig

Type the name of your system using alphanumeric characters.
The first character must be a letter.  For example, tinker.

Type your system name: tinker

You typed tinker as the name of your system.
Is this correct? Type y or n [y]: y

A system with that name already exists. Replace it? (y/n) [y]: y

*** SPECIFY THE DATE AND TIME ***

Enter the current date and time in this format:
yymmddhhmm. Use two digits for year (yy),
month (mm), day (dd), hour (hh), and minute (mm).
You type the time in 24-hour format. For example,
for 11:30 p.m. on May 3, 1990, the response
would be:

     9005032330

Type the date and time [no default]: 9005032330

*** SPECIFY THE TIME ZONE INFORMATION ***

Enter the time zone for your area, using the options
listed in the table below.  You can also enter the number of
hours (-12 to 12) in time east of Greenwich.

  Selection     Time Zone
---------------------------------
      e         Eastern
      c         Central
      m         Mountain
      p         Pacific
      g         Greenwich
---------------------------------

Enter your choice: p


Does your area alternate between Daylight Savings
and Standard time [yes] ? yes


Select your geographic area for Daylight Savings Time,
using the options in the table below.

  Selection     Geographic Area
--------------------------------
```

Example 2-1: (continued)

```
      u          USA
      a          Australia
      e          Eastern Europe
      c          Central Europe
      w          Western Europe
--------------------------------

Enter your choice [u]: u

Thurs May 10 12:29:00 EDT 1990

*** System Configuration Procedure ***


Configuration file complete.

Do you want to edit the configuration file? (y/n) [n]: y

      .
      .  < You would be editing here >
      .


*** PERFORMING SYSTEM CONFIGURATION ***
    working ..... Sun May 13 09:40:44 EDT 1990
    working ..... Sun May 13 09:42:45 EDT 1990

*** DEVICE SPECIAL FILE CREATION ***
    working ..... Sun May 13 09:44:08 EDT 1990

A log file listing Special Device Files is located in /dev/MAKEDEV.log

The new kernel is /sys/VAX/TINKER/vmunix
```

2.2.2 Testing the New Kernel

On completion of the automatic configuration process, you can test the new kernel that you have built by performing the following steps:

1. Put the newly created kernel in the root directory. For instance, to put the kernel created in Example 2-1 into the root directory, you would type:

   ```
   # mv /sys/VAX/TINKER/vmunix /vmunix
   # chmod 755 /vmunix
   ```

2. Reboot the system:

   ```
   # /etc/reboot
   ```

If you have problems booting your new kernel, you may have made errors in your configuration file. You can use the original kernel you copied to `/sys/vmunix.old` while you correct any errors in your new configuration file. Refer to Section 2.5 for instructions.

2.3 Building a New Kernel Manually

You can build a new kernel manually in either single-user or multi-user mode. However, it is recommended that you build it in single-user mode, so the build process is protected from users.

You can shut down the system to single-user mode with the following command:

```
# /etc/shutdown +5 "Building a new kernel"
```

To build a new kernel manually in either single-user or multi-user mode, you must perform the following steps:

1. Edit the configuration file.
2. Run the `config` utility.
3. Define code dependencies.
4. Compile and load the binary files.
5. Boot the new kernel.

Each of these steps is described in the following sections. You must follow these steps consecutively.

2.3.1 Edit the Configuration File

The configuration file resides in one of the following two directories, depending on what type of hardware you have:

- The `/sys/conf/vax` directory
- The `/sys/conf/mips` directory

The configuration file has the same name as your system, but in uppercase letters. For example, if your VAX system is named `myvax`, your configuration file is named `/sys/conf/vax/MYVAX`. If your RISC system is named `mymips`, your configuration file is named `/sys/conf/mips/MYMIPS`.

The configuration file is the file you copy and edit when you build a new kernel. This file includes definitions for all supported devices. The supported devices are listed in Appendix A.

Follow these steps to copy and then to edit the configuration file:

1. Log in to the system as superuser (`root`).
2. Change your working directory to `/sys/conf/vax` or `/sys/conf/mips` by typing one of the following commands:

   ```
   # cd /sys/conf/vax
   # cd /sys/conf/mips
   ```

3. Make a backup copy of the original configuration file. To do this, copy the original configuration file to another file in the same directory.

   For example, if your configuration file is `MYVAX`, type the following:

   ```
   # cp MYVAX MYVAX.old
   ```

   If your configuration file is `MYMIPS`, type the following:

   ```
   # cp MYMIPS MYMIPS.old
   ```

4. Change the file access permissions (mode) of the working configuration file to permit the owner to write to it. For example, if your working configuration file is named `MYVAX`, type the following:

   ```
   # chmod +w MYVAX
   ```

If your working configuration file is named MYMIPS, type the following:

```
# chmod +w MYMIPS
```

5. Edit the working file. Use a text editor, such as the vi editor, to add or delete entries in the MYVAX or MYMIPS working configuration file. Use the format and rules described in Chapter 1 to make changes to the configuration file.

2.3.2 Run the config Utility

When you have edited the configuration file, run the config utility to create directories in which to store binary files.

Follow these steps to generate the new directories:

1. Make sure that your working directory is either /sys/conf/vax or /sys/conf/mips. (You should be in this directory after editing the configuration file.)
2. Run the config utility with the name of the working configuration file you edited in Section 2.3.1. For example, if your configuration file is named MYVAX, issue the following command:

```
# config MYVAX

Don't forget to run "make depend"
```

If your configuration file is named MYMIPS, issue the following command:

```
# config MYMIPS

Don't forget to run "make depend"
```

The utility creates a directory with the same name as your configuration file, if it does not already exist. For example, if your system is a VAX system and your configuration file is named MYVAX, the config utility creates the directory /sys/VAX/MYVAX. If your system is a RISC system and your configuration file is named MYMIPS, the config utility creates the directory /sys/MIPS/MYMIPS. When the utility finishes creating the directory, it displays a message to remind you to execute the make command with the depend parameter. For more information, see make(1) in the *ULTRIX Reference Pages*.

2.3.3 Define the Code Dependencies

Your next step is to define the code dependencies. The code dependencies determine which binary files are needed and how they are built, based on the configuration of your kernel.

To define the code dependencies:

1. Change your working directory to directory config created in Section 2.3.2. For example, if your system configuration file is named MYVAX, issue the following command:

```
# cd /sys/VAX/MYVAX
```

If your system configuration file is named MYMIPS, issue the following command:

```
# cd /sys/MIPS/MYMIPS
```

2. Execute the make command with the clean parameter. The following example shows how to issue this command:

```
#  make clean
```

This command ensures that the /sys/VAX/MYVAX directory or the /sys/MIPS/MYMIPS directory contains only the required files for creating the kernel specified by the MYVAX or the MYMIPS configuration file.

3. Execute the make command with the depend parameter, as shown in the following example:

```
# make depend
```

This command instructs make to build or rebuild the rules that it needs to recognize interdependencies in the system source code. Executing this command ensures that any changes to the system source code will be recompiled the next time you run the make command. The make command modifies the makefile, appending the dependencies to the end of the file. After make successfully completes, it updates the makefile.

2.3.4 Compile and Load the Binary Files

After defining the code dependencies, compile and load the new binary files, using the makefile that you just created.

To compile and load the binary files:

1. Use the make command to produce a complete binary system image, the kernel. The kernel is stored in the current directory. The system responds by displaying a number of messages as it compiles and loads the binary files. When the make command completes, the system redisplays the system prompt.

   The following example shows how you issue the make command (the output from the command may be different from what is shown here):

```
# make
/bin/rm -f a.out a.out.q assym.h
      .
      .
      .
#
```

2. If the system is in multi-user mode, you must now shut it down to single-user mode, by typing:

```
# /etc/shutdown +5 "Building a new kernel"
```

3. Because you may have made errors in your configuration file, you should save the original kernel. If the new kernel fails, you can recover by booting from the generic kernel, /genvmunix, and correct any errors in your configuration file. Move the original kernel to another filename. The following example shows how to move the kernel:

```
# mv /vmunix  /sys/vmunix.old
```

4. The output of the make command is a kernel named vmunix in the current directory. Move this file to the root directory and then change its mode. For example:

```
# mv vmunix  /vmunix
# chmod 755 /vmunix
```

The original /vmunix file is replaced by the new vmunix file and is ready to be booted. The original /vmunix resides in /sys/vmunix.old because you copied it there in step 3.

2.3.5 Boot the New Kernel

Use the reboot command to boot the new kernel, /vmunix. To boot the new kernel, type:

```
# /etc/reboot
```

In this example, the processor halts and then automatically reboots using the default boot device. The system boots the /vmunix image.

If the new kernel fails to boot or displays errors, you can recover by booting the original kernel, vmunix.old, and running that kernel until you determine the cause of the problem. Refer to Section 2.5 for instructions.

2.4 Building a Kernel After a Capacity Upgrade Installation

If you installed a larger user capacity License Management Facility (LMF) key, or plan to use a capacity upgrade, you may need to increase the maximum number of users to match this capacity, and then build a new kernel.

The maxusers parameter in the configuration file should match the number of authorized users in your capacity upgrade installation kit or in your License Management Facility PAK. If your capacity is unlimited, then maxusers should match the maximum number of simultaneous user logins.

To determine the current value of maxusers, type the following:

```
# grep maxusers /sys/conf/{vax,mips}/HOSTNAME
```

Use the following procedure to increase maxusers and build a new kernel (if necessary):

1. Log is as superuser (root).
2. Set the EDITOR environment variable to specify the text editor you want to use to edit the configuration file. For example, to use the ex editor, type:

   ```
   # EDITOR=ex
   # export EDITOR
   ```

3. Execute the doconfig program with the -c option to build a new kernel from your existing configuration file:

   ```
   # /etc/doconfig -c HOSTNAME
   ```

4. Type yes when the doconfig program asks if you want to edit the configuration file. The program ecapes you to your editor. You then change the maxusers parameter to the new number of authorized users. For example, if you have an upgrade installation kit for 64 users the new entry would read:

   ```
   maxusers   64
   ```

5. Exit from the editor; the doconfig program then builds the new kernel.
6. Shut the system down to single-user mode:

   ```
   # /etc/shutdown +5 "Installing new kernel"
   ```

7. Save the running vmunix kernel as vmunix.old by typing:

```
# mv /vmunix /sys/vmunix.old
```

8. Put the newly created kernel into the root directory:

```
# mv /sys/{VAX,MIPS}/HOSTNAME/vmunix /vmunix
# chmod 755 /vmunix
```

9. Reboot the system:

```
# /etc/reboot
```

If you have problems booting the new kernel, refer to Section 2.5 for instructions on how to recover your original kernel.

2.5 How to Recover When a New Kernel Fails to Boot

If you have problems booting your new kernel, use the following procedure to recover the original kernel, vmunix.old:

1. Boot the generic kernel to single-user mode. Refer to the *Guide to Shutdown and Startup* for instructions on how to boot your processor. You use a conversational mode boot to boot the generic kernel /genvmunix.

2. Check your file systems:

```
# /etc/fsck -p
```

3. Mount your local file systems:

```
# /etc/mount -a -t ufs
```

4. Copy the original kernel to the root directory:

```
# cp /sys/vmunix.old /vmunix
```

5. Reboot the system:

```
# /etc/reboot
```

Device Mnemonics A

This appendix breifly lists and describes the mnemonics that are used to attach hardware and software to your system during configuration. For each device attached to your system, there is a corresponding special file which is created by the `MAKEDEV` shell script.

Section 4 of the *ULTRIX Reference Pages* provides detailed information on the use of each mnemonic in relation to both the system configuration file and the `MAKEDEV` shell script. For more information on the `MAKEDEV` script, see the `MAKDEV`(8) reference page. If online reference pages are available, you can use the `man` command to display a specific mnemonic. For example, to display the Mass Storage Control Protocol (MSCP) disk controller driver, type:

```
% man ra
```

If applicable, the **Syntax** section of the reference page displays the syntax of the device as it should appear in the system configuration file.

Table A–1 divides the list of mnemonics into nine categories: generic, systems, consoles, disks, tapes, terminals, modems, printers, and others. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The systems category lists the mnemonic for the DECstation 3100 system setup. The consoles category lists the system console devices that the ULTRIX operating system uses. The disks, tapes, terminals, modems, and printers categories identify the appropriate mnemonics for those devices. The others category lists the mnemonic for DECstation 3100 devices.

In Table A–1, some mnemonics are followed by an asterisk (*) and some device names are followed by one or more question marks (?). The asterisk (*) and question mark (?) represent a variable number. For example, `ra*` refers to all the ra devices associated with the MSCP disk controller. Likewise, `8??0` refers to the VAX 8000 series including the 8200, 8300, 8500, 8550, and so on.

Table A-1: Devices Supported by MAKEDEV

Category	Mnemonic	Description
Generic	boot*	Boot and std devices by cpu number; for example, boot750
	mvax*	All MicroVAX setups; for example, mvax2000
	vaxstation*	A VAXstation 2000 setup; for example, vaxstation2000
	std	Standard devices with all console subsystems:
	drum	Kernel drum device
	errlog	Error log device
	audit	Audit log device
	kUmem	Kernel Unibus/Q-bus virtual memory
	kmem	Virtual main memory
	mem	Physical memory
	null	A null device
	trace	A trace device
	tty	A character terminal device
	local	Customer-specific devices
Systems	DECstation	A DECstation 3100 setup
Consoles	console	System console interface
	crl	Console RL02 disk interface for VAX 86?0
	cs*	Console RX50 floppy interface for VAX 8??0
	ctu*	Console TU58 cassette interface for VAX 11/725/730/750
	cty*	Console extra serial line units for VAX 8??0
	cfl	Console RX01 floppy interface for 11/78?
	ttycp	Console line used as auxiliary terminal port
Disks	hp*	MASSBUS disk interface for RM?? drives and RP?? devices
	ra*	UNIBUS/Q-bus/BI/HSC/DSSI MSCP disk controller interface
	rb*	UNIBUS IDC RL02 disk controller interface for RB?? drives
	rd*	VAXstation 2000 and MicroVAX 2000 RD type drives
	rz	SCSI disks (RZ22/RZ23/RZ55/RRD40)
	rk*	UNIBUS RK?? disk controller interface
	rl*	UNIBUS/Q-bus RL?? disk controller interface
	rx*	VAXstation 2000 and MicroVAX 2000 RX type drives
Tapes	mu*	TU78 MASSBUS magtape interface
	tms*	UNIBUS/Q-bus/BI/HSC/DSSI TMSCP tape controller interface
	rv*	UNIBUS/Q-bus/BI TMSCP optical disk
	ts*	UNIBUS/Q-bus TS11/TS05/TU80 magtape interface
	tu*	TE16/TU45/TU77 MASSBUS magtape interface
	st*	VAXstation 2000 and MicroVAX 2000 TZK50 cartridge tape
	tz*	SCSI tapes (TZ30/TZK50)
Terminals	cxa*	Q–bus cxa16
	cxb*	Q–bus cxb16
	cxy*	Q–bus cxt08
	dfa*	Q–bus DFA01 comm multiplexer
	dhq*	Q–bus DHQ11 comm multiplexer
	dhu*	UNIBUS DHU11 comm multiplexer

Table A-1: (continued)

Category	Mnemonic	Description
	dhv*	Q–bus DHV11 comm multiplexer
	dmb*	BI DMB32 comm multiplexer including dmbsp serial printer/plotter
	dhb*	BI DHB32 comm multiplexer
	dmf*	UNIBUS DMF32 comm multiplexer including dmfsp serial printer/plotter
	dmz*	UNIBUS DMZ32 comm multiplexer
	dz	UNIBUS DZ11 and DZ32 comm multiplexer
	sh*	MicroVAX 2000, 8 serial line expansion option
	ss*	VAXstation 2000 and MicroVAX 2000 basic 4 serial line unit
	fc*	VAXstation 60 basic 4 serial line unit
	dzq*	Q–bus DZQ11 comm multiplexer
	dzv*	Q–bus DZV11 comm multiplexer
	lta*	Sets of 16 network local area terminals (LAT)
	pty*	Sets of 16 network pseudoterminals
	qd*	Q–bus VCB02 (QDSS) graphics controller/console
	qv*	Q–bus VCB01 (QVSS) graphics controller/console
	sm*	VAXstation 2000 monochrome bitmap graphics/console
	sg*	VAXstation 2000 color bitmap graphics console
	lx	VAXstation 8000 color high-performance 3D graphics
	fg*	VAXstation 60 color bitmap graphics/console
Modems	dfa*	DFA01 integral modem communications device.
Printers	dmbsp*	BI DMB32 serial printer/plotter
	dmfsp*	UNIBUS DMF32 serial printer/plotter
	lp*	UNIBUS LP11 parallel line printer
	lpv*	Q–bus LP11 parallel line printer
Packet filter	pfilt	Packet filter devices; set of 64
Other	pm*	mono/color bitmap graphics/mouse/modem /printer/terminals for DECstation 3100

Index

D

N

P

Q

R

S

T

U

V

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to Configuration File Maintenance
AA-ME90C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? ______________________________

What do you like best about this manual? ______________________________

What do you like least about this manual? ______________________________

Please list errors you have found in this manual:

Page	Description
______	______________________________
______	______________________________
______	______________________________
______	______________________________
______	______________________________

Additional comments or suggestions to improve this manual:

____

What version of the software described by this manual are you using? ____________

Name/Title ______________________________ Dept. ____________
Company ______________________________ Date ____________
Mailing Address ______________________________
______________ Email ______________ Phone ____________

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–3/Y32
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to Configuration File Maintenance
AA-ME90C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? ______________________________

____

____

What do you like best about this manual? ______________________________

____

____

What do you like least about this manual? ______________________________

____

____

Please list errors you have found in this manual:

Page	Description
______	____
______	____
______	____
______	____
______	____

Additional comments or suggestions to improve this manual:

____

____

____

____

What version of the software described by this manual are you using? ______________

Name/Title ______________________________________ Dept. ______________

Company ____ Date ______________

Mailing Address ____

______________________ Email ______________________ Phone ______________

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–3/Y32
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

digital